

UNIVERSIDADE NOVA DE LISBOA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica e de Computadores

Melhoria de Protocolos de Encaminhamento em VANETs de Alta Densidade

Por
Nuno Miguel Abreu Luís

Dissertação apresentada na Faculdade de
Ciências e Tecnologia da Universidade Nova de
Lisboa para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores.

Orientação Científica: Prof. Doutor Rodolfo Oliveira
Prof. Doutor Luís Bernardo

Lisboa
2009

Agradecimentos

Gostaria de apresentar os meus agradecimentos a todas as pessoas que de certo modo me ajudaram na realização desta dissertação.

Agradeço ao Prof. Rodolfo Oliveira, que mostrou ser um excelente professor e orientador, pelo ensino, pela sua disponibilidade e também pela preciosa ajuda na realização e orientação desta dissertação. Esteve presente sempre que necessitei, mesmo em horários mais complicados e em momentos de muito trabalho. Agradeço também pela rigorosa e cuidada revisão da escrita desta dissertação, e o apoio demonstrado na elaboração do artigo publicado.

Agradeço também ao Prof. Luís Bernardo pela co-orientação, ensino e apoio demonstrado ao longo da realização do projecto desta dissertação, assim como na verificação rigorosa e cuidada da escrita da dissertação e do artigo publicado.

A todos os meus colegas da FCT-UNL, que conviveram comigo ao longo destes anos e que sempre me apoiaram, durante a realização do meu curso. Agradeço a todos os que comigo passaram longas noites a estudar no segundo piso do departamento de engenharia electrotécnica, destacando os meus amigos e *Companheiros de Barco*, Pedro Domingos, Carlos Dias e Filipe Brazão, pela partilha de todos os momentos, bons e menos bons, vividos durante este ciclo.

Aos meus colegas de gabinete e amigos Francisco Ganhão, Miguel Pereira, Michel Rodrigues, Michael Figueiredo, Edinei Santin, Cláudio Assunção e Miguel Luzio pela com-

panhia, apoio e esclarecimento em diversas alturas da implementação do projecto desta dissertação.

Aos meus amigos João Simões, Filipa Silva, Inês Freire, Marta Soares, Sérgio Rodriguez e André Falcão pelos longos anos de convívio e experiências partilhadas.

Por fim, mas de forma alguma com menor importância, gostaria de agradecer à minha família por todo o carinho e apoio demonstrado ao longo destes anos. Agradeço aos meus pais, à minha irmã, madrinha e avô que sempre me souberam motivar e incentivar ao longo da minha vida de estudante, e que sem eles, a realização deste curso não seria possível. À Patrícia Duarte pela disponibilidade, apoio, carinho e compreensão por todos momentos que passei privado da sua companhia.

Obrigado.

Sumário

A elevada mobilidade das redes *ad-hoc* veiculares influencia significativamente o comportamento dos protocolos utilizados. Muitos dos protocolos de rede desenvolvidos para redes *ad-hoc* móveis exibem fraco desempenho em redes veiculares, dadas as suas características de alta mobilidade e restrição de movimentos. Como tal, este trabalho apresenta um método de controlo de topologia para redes *ad-hoc* veiculares. O controlo de topologia pretende caracterizar as relações de comunicação entre dois nós, identificando as ligações que apresentam uma duração mais prolongada. Este tipo de ligações entre nós oferece maior estabilidade aos protocolos de encaminhamento, diminuindo simultaneamente o tráfego de controlo. As ligações de maior duração são identificadas por um algoritmo de baixa complexidade, sendo um critério para realizar o agrupamento de nós de forma a diminuir o tráfego de controlo associado às operações de inundação da rede.

A utilidade da criação de grupos com base na duração das ligações é demonstrada através da integração do algoritmo no protocolo de encaminhamento OLSR. Comparam-se as estatísticas da taxa de sucesso de encaminhamento e do tempo necessário a visitar o nó de destino (atraso do caminho), caso exista caminho para esse nó. Os resultados obtidos com as propostas descritas nesta dissertação exibem uma melhoria de desempenho, quando comparados ao protocolo de encaminhamento OLSR. Este facto traduz-se num aumento da taxa de sucesso de encaminhamento e na redução do atraso do caminho, sendo mais significativo em cenários onde a densidade de nós é mais elevada.

Palavras Chave: Controlo de Topologia, Protocolos de Encaminhamento, Redes *ad-hoc* Veiculares.

Abstract

In mobile networks, particularly in vehicular ad-hoc networks (VANETs), the topology is highly dynamic due to the movement of the nodes, hence an on-going session suffers frequent path breaks. This work presents a method to control the network's topology. The available knowledge about the stability of the network's topology is used to improve the routing protocol's performance through decreasing the probability of path breaks. Long-duration links provide greater stability to the routing protocols, and can be simultaneously used for decreasing the amount of traffic control. The links with a longer duration are identified by a low complexity algorithm. This criteria may be used to define groups of nodes, decreasing the traffic control associated to flooding operations.

The usefulness of creating groups based on link duration is demonstrated through the integration of the algorithm in the OLSR routing protocol. This proposal is compared with the original OLSR protocol in terms of packet delivery success rate and end-to-end delay. The results confirm the success of the proposed topology control algorithm, exhibiting higher packet delivery rate and lower end-to-end delay than the original OLSR protocol. These improvements are even more significant for higher node densities.

Keywords: Topology Control, Routing Protocols, Vehicular ad-hoc Networks.

Acrónimos

A-STAR Anchor-based Street and Traffic Aware Routing

AODV Ad hoc On Demand Distance Vector

ASDM Adaptive Space Division Multiplexing

BROADCOMM BROADcast COMMunications

CBLR Cluster-Based Location Routing algorithm

COIN Clustering for Open IVC Networks

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTS Clear To Send

DARPA Defense Advanced Research Projects Agency

DCF Distributed Coordination Function

DRG Distributed Robust Geocast

DSDV Destination Sequenced Distance Vector

DSR Dynamic Source Routing

GB Grupo de Broadcast

GeOpps Geographical Opportunistic routing for vehicular networks

GPS Global Positioning System

GPSR Greedy Perimeter Stateless Routing

GSR Global State Routing

IDM-IC Intelligent Driver Model with Lane Changes

IDM-IM Intelligent Driver Model with Intersection Management

IEEE Institute of Electrical and Electronics Engineers

ITS Intelligent Transportation System

IVC Inter-Vehicle Communications

LAN Local Area Network

LGB Líder de Grupo de Broadcast

LREP Location REPLY

LREQ Location REQuest

MAC Medium Access Control

MACAW Multiple Access with Collision Avoidance for Wireless

MANET Mobile Ad-hoc NETwork

MCDS Minimum Connected Dominating Set

MOVE The Mobility Model Generator for Vehicular Networks

MPR Multipoint Relay

NAV Network Allocation Vector

OFDM Orthogonal Frequency-Division Multiplexing

OLSR Optimized Link State Routing

OSI Open Systems Interconnection Reference Model

PCF Point Coordination Function

PRAODV PReemptive AODV

PRAODV-M PReemptive AODV-Maximum

ROVER RObust VEhicular Routing

RTS Request To Send

SDMA Space-Division Multiple Access

STRAW STreet RAndom Waypoint

SUMO Simulation of Urban MObility

SWANS Scalable Wireless Ad Hoc Network Simulator

TC Topology Control

TDM Time Division Multiplexing

TraNS Traffic and Network Simulation environment

V-PEACE Vehicle Position Environment Acquisition and Communication Evolution

VC-MAC VehiCular-MAC

V2V Vehicle-to-Vehicle

VANET Vehicular Ad-hoc NETwork

Veins Vehicles in network simulation

WAVE Wireless Access in Vehicular Environments

WIS Weighted Independent Set

WLAN Wireless LAN

WTRP Wireless Token Ring Protocol

ZOR Zone of Relevance

Conteúdo

1	Introdução	1
1.1	Introdução	1
1.2	Motivação	2
1.3	Objectivos e Contribuições	4
1.4	Estrutura da Dissertação	4
2	Redes Veiculares	7
2.1	Introdução	7
2.2	Sub-camada MAC	10
2.2.1	Métodos de Alocação Estática de Recursos	11
2.2.2	Métodos de Alocação Dinâmica de Recursos	12
2.3	Protocolos de Encaminhamento	18
2.3.1	Encaminhamento <i>ad-hoc</i>	19
2.3.2	Encaminhamento Baseado na Localização	21
2.3.3	Encaminhamento Baseado em <i>Clusters</i>	23
2.3.4	Encaminhamento por <i>Broadcast</i>	25
2.3.5	Encaminhamento <i>Geocast</i>	26
2.3.6	Comparação Entre Protocolos de Encaminhamento	28
3	Modelos de Mobilidade	31
3.1	Introdução	31
3.2	Aplicações Geradoras de Cenários de Mobilidade	36
3.3	Cenários de Mobilidade Gerados	39
4	Identificação de Nós Âncora	45
4.1	Introdução	46
4.2	Identificação de Ligações Consoante a Duração	47
4.3	Seleccção de Nós Âncora	50
4.4	Avaliação de Desempenho	57
5	Encaminhamento com Optimização de Topologia	65
5.1	<i>Optimized Link State Routing</i> (OLSR)	66

5.1.1	Visão Global	66
5.1.2	Funcionamento do Protocolo	68
5.2	Modificações Propostas	78
6	Análise de Desempenho	85
6.1	Descrição da Simulação	85
6.2	Método de Recolha de Dados	88
6.3	Resultados	89
7	Conclusões	101
7.1	Considerações Finais	101
7.2	Trabalho Futuro	102
A	Aplicação Auxiliar para a Ferramenta SUMO	105
B	<i>Script</i> de Simulação do <i>ns-2</i>	113
C	Publicações	119
	Bibliografia	132

Lista de Figuras

1.1	Desempenho do protocolo OLSR, relativamente à percentagem de sucesso de pedidos de encaminhamento (a) e do tempo médio do caminho (b), num cenário de auto-estrada, na presença de tráfego em apenas um, e nos dois sentidos.	3
2.1	Arquitecturas de redes veiculares.	9
2.2	Funcionamento do protocolo CSMA/CA.	14
2.3	Esquema de funcionamento do protocolo V-PEACE.	16
2.4	Funcionamento de transmissão do protocolo WTRP.	17
2.5	Troca de pacotes do protocolo VC-MAC.	18
2.6	Exemplo de encaminhamento <i>greedy</i>	22
2.7	Formação de <i>clusters</i> numa rede veicular.	24
2.8	Constituição de células utilizado pelo BROADCAST.	27
2.9	Constituição da ZOR em protocolos de encaminhamento <i>geocast</i>	27
3.1	Duração das ligações físicas dos cenários Cen_{4v} (a), Cen_{6v} (b), Cen_{8v} (c) e Cen_{10v} (d).	43
3.2	Função de distribuição acumulada da duração das ligações físicas dos 4 cenários de simulação.	44
4.1	Rede <i>ad hoc</i> constituída por 6 nós ($\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$).	49
4.2	Rede <i>ad hoc</i> constituída por 6 nós ($\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$), onde existem 3 GB, e os nós n_1, n_5 e n_6 são LGB.	52
4.3	Rede móvel <i>ad hoc</i> constituída por 6 nós ($\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$) em diferentes instantes temporais: (a) os nós n_2 e n_4 não possuem ligações lógicas; (b) os nós n_2 e n_4 possuem ligações lógicas com os nós n_3 e n_6 , e n_3 e n_5 respectivamente.	55
4.4	Árvore de grupos de <i>broadcast</i> do nó n_1 representado no exemplo da Figura 4.3(b).	56
4.5	Exemplo de fusão de um GB.	56

4.6	Árvore de grupos de <i>broadcast</i> no caso pior em que a conectividade entre os GB $\{n_1, n_3, n_5, n_6\}$ e $\{n_7, n_8\}$ são separados por dois nós não LGB (nós n_3 e n_7).	57
4.7	Tempo médio de eleição do LGB para os cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .	58
4.8	Número médio de vizinhos por nó (teórico e real), e dimensão média do GB obtida com o Algoritmo 4.1 aplicado ao cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .	59
4.9	Percentagem das durações da estabilidade e da instabilidade em tempo por nó nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .	60
4.10	Número médio de GB nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .	61
4.11	Comparação entre as durações das ligações físicas e ligações lógicas dos cenários Cen_{4v} (a), Cen_{6v} (b), Cen_{8v} (c) e Cen_{10v} (d).	62
5.1	Rede sem fios com a utilização de nós MPR.	68
5.2	Elaboração de uma rota através de informação existente na tabela de topologia.	75
6.1	Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), do protocolo OLSR, aplicado aos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .	89
6.2	Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolo OLSR e OLSR-FCT, aplicado ao cenário Cen_{4v} , para diferentes valores de cobertura.	90
6.3	Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolo OLSR e OLSR-FCT, aplicado ao cenário Cen_{6v} , para diferentes valores de cobertura.	91
6.4	Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolo OLSR e OLSR-FCT, aplicado ao cenário Cen_{8v} , para diferentes valores de cobertura.	92
6.5	Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolo OLSR e OLSR-FCT, aplicado ao cenário Cen_{10v} , para diferentes valores de cobertura.	93
6.6	Comparação do desempenho do protocolo OLSR-FCT, relativamente à percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), nos quatro cenários de simulação.	94
6.7	Influência do limiar da estabilidade (k_{est}) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 6 vizinhos (Cen_{6v}), com uma cobertura de 85%.	95
6.8	Influência do <i>timeout</i> do <i>beacon</i> (T_O) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 6 vizinhos (Cen_{6v}).	96

6.9	Influência do <i>timeout</i> do <i>beacon</i> (T_O) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 8 vizinhos (Cen_{8v}).	97
6.10	Influência do <i>timeout</i> do <i>beacon</i> (T_O) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 10 vizinhos (Cen_{10v}).	97
6.11	Comparação do desempenho do protocolo OLSR na presença de trânsito em um sentido, nos dois sentido e do protocolo OLSR-FCT com trânsito nos dois sentidos.	98
A.1	Ilustração dos conceitos de <i>node</i> , <i>edge</i> e <i>route</i> , utilizados pela ferramenta SUMO.	106

Lista de Tabelas

2.1	Algoritmos de encaminhamento aplicados a redes veiculares.	29
3.1	Modelos de mobilidade.	36
3.2	Caracterização das classes de veículos implementadas.	41
4.1	Exemplo do conteúdo da tabela de <i>beacons</i> do nó n_3 representado na Figura 4.1 no instante $t=102.5s$, sendo $T_B = 1s$	50
4.2	Exemplo do conteúdo da tabela de <i>beacons</i> do nó n_3 representado na Figura 4.2 no instante $t=102.5s$, sendo $T_B = 1s$, após eleição de nós LGB.	53
4.3	Duração média de actividade de um nó, nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v}	58
6.1	Número médio total de pedidos de encaminhamento realizados durante uma simulação, nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v}	86
6.2	Valores de T_O nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v}	87

Capítulo 1

Introdução

1.1 Introdução

Nos dias de hoje, o automóvel é o meio de transporte mais utilizado por milhões de pessoas no seu quotidiano. Com a crescente utilização deste meio de transporte, adveio a necessidade de possibilitar a comunicação entre viaturas, com o intuito de fornecer segurança e entretenimento aos seus ocupantes.

A excessiva utilização do automóvel tem contribuído, não só para a criação de situações de saturação de tráfego, mas também para o crescente número de situações de perigo, que têm vindo a aumentar consideravelmente a probabilidade de ocorrência de acidentes rodoviários. Estas razões têm motivado a implementação de aplicações que possam auxiliar o condutor da viatura, com o intuito de melhorar a segurança dos seus ocupantes, assim como auxiliá-lo no planeamento de trajectos (de modo a evitar congestionamentos), ou até aumentar o grau de automação da viatura.

No campo do entretenimento, a existência de um sistema de comunicações entre viaturas pode trazer grandes vantagens para os seus ocupantes, permitindo a partilha de músicas e vídeos, ou até mesmo a interacção entre os ocupantes de diferentes veículos, particularmente útil em viagens de longa duração.

Uma das soluções capaz de promover a comunicação entre veículos, principalmente in-

dicada para o tema do entretenimento, passa pela criação de uma rede *ad-hoc* veicular (*Vehicular ad-hoc Network* - VANET). Uma rede *ad-hoc* caracteriza-se por ser uma rede em que todos os nós que a compõem funcionam como encaminhadores, não sendo necessário recorrer a uma infra-estrutura fixa (normalmente criada através de *access points*, ou estações base). As redes *ad-hoc* são utilizadas, quando não existe uma infra-estrutura fixa que permita ligar os dispositivos, e beneficiam de uma fácil configuração. No entanto, a maioria das redes *ad-hoc* caracterizam-se por serem redes estáticas ou de mobilidade moderada (*Mobile Ad-hoc Network* - MANET).

As VANETs podem ser caracterizadas como um caso particular das redes *ad-hoc* móveis, diferenciadas por exemplo, no facto da mobilidade dos nós que compõem uma VANET ser geralmente maior, e de estar limitada pela orientação das estradas. Como tal, o problema na criação de uma rede *ad-hoc* veicular está no desenho dos protocolos de encaminhamento utilizados neste tipo de redes, e no desempenho que estes possam apresentar em situações de redes veiculares.

1.2 Motivação

Actualmente, o protocolo de encaminhamento mais utilizado em redes *ad-hoc* é o *Optimized Link State Routing* (OLSR) [JMC⁺01]. Um protocolo com optimização de topologia, como o OLSR, tem um modo de funcionamento bastante característico: cada nó, selecciona de entre os seus nós vizinhos, uma quantidade de nós suficientes de modo a cobrir toda a vizinhança a dois saltos do nó, com o intuito de realizar o encaminhamento e difundir as mensagens topológicas utilizadas neste protocolo. Os nós eleitos são designados de nós MPR (*Multipoint Relays*), ou ainda nós âncora. No funcionamento de protocolos aplicados a redes *ad-hoc*, a grande vantagem da utilização deste tipo de nós, está na capacidade de reduzir a quantidade de tráfego de controlo gerado, transmitido em modo *broadcast*.

Tendo em conta as características apresentadas anteriormente, e por ser dos protocolos mais utilizados, realizaram-se vários testes com o intuito de avaliar o desempenho do protocolo OLSR num cenário de redes veiculares. Na Figura 1.1, é apresentado o desempenho

do protocolo de encaminhamento OLSR num cenário de auto-estrada, com dois sentidos, e três faixas por sentido. A densidade média é de 6 vizinhos por veículo, em cada sentido. Como se pode observar pela Figura 1.1(a), o protocolo OLSR apresenta uma percentagem de sucesso de pedidos de encaminhamento de quase 70% quando existe tráfego em apenas um sentido. No entanto, quando é adicionado tráfego a viajar em sentido contrário, a taxa de sucesso decresce até 48%, verificando-se uma forte degradação do desempenho do protocolo.

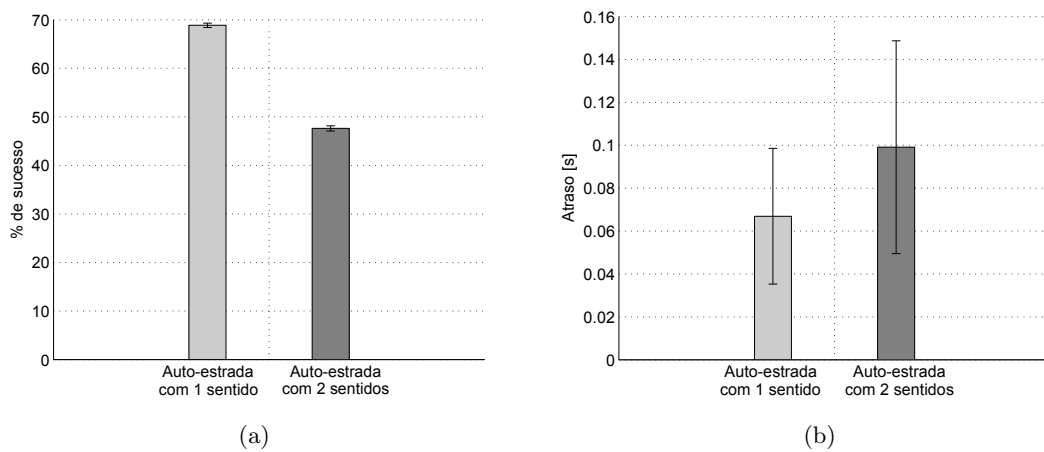


Figura 1.1: Desempenho do protocolo OLSR, relativamente à percentagem de sucesso de pedidos de encaminhamento (a) e do tempo médio do caminho (b), num cenário de auto-estrada, na presença de tráfego em apenas um, e nos dois sentidos.

Relativamente ao atraso médio dos caminhos, ilustrado na Figura 1.1(b), na presença de tráfego em ambos os sentidos, o desempenho do protocolo OLSR diminui, pois o atraso médio exibe um aumento relativo de aproximadamente 30%, passando dos 70 *milissegundos* para os 100 *milissegundos*. Pode-se então concluir que, tendo em conta o exemplo apresentado, o desempenho do protocolo OLSR é fortemente influenciado pela mobilidade (mais concretamente pela existência de nós a circular em sentido contrário). Este facto motiva todo o trabalho apresentado na dissertação, o qual pretende solucionar o problema descrito anteriormente.

1.3 Objectivos e Contribuições

Esta dissertação tem como principal objectivo melhorar o desempenho do protocolo de encaminhamento OLSR num cenário de redes veiculares, mais especificamente em cenários de auto-estrada, tentando anular a degradação do mesmo observada na secção anterior. Para tal, pretende-se utilizar um algoritmo capaz de identificar as ligações existentes entre veículos que viajam no mesmo sentido, caracterizadas por serem as ligações com maior duração. Identificadas estas ligações, pretende-se modificar o método de eleição de nós âncora utilizado no funcionamento do protocolo OLSR, realizando uma eleição baseada na estabilidade das ligações existentes entre os veículos. Desta forma, espera-se que o desempenho do OLSR deixe de ser prejudicado em cenários de redes veiculares, com a presença de trânsito em ambos os sentidos.

O trabalho descrito nesta dissertação originou uma publicação numa conferência nacional [LOBP09], estando outro artigo em fase de revisão numa conferência internacional. Estes artigos estão incluídos no Apêndice C.

1.4 Estrutura da Dissertação

A dissertação encontra-se dividida em sete capítulos e três apêndices. No Capítulo 2 ("Redes Veiculares") é realizado um levantamento geral sobre o estado da arte na área de redes veiculares. São apresentadas as características de uma rede veicular, assim como alguns desenvolvimentos realizados na sub-camada de Controlo de Acesso ao Meio e ao nível dos protocolos de encaminhamento.

No Capítulo 3 ("Modelos de Mobilidade") é feita uma revisão dos modelos de mobilidade existentes, sendo também apresentadas algumas aplicações geradoras de modelos de mobilidade, com especial atenção para a aplicação utilizada durante o projecto desta dissertação. Por fim, no final do capítulo, são apresentados os cenários de simulação utilizados para avaliar o desempenho do protocolo OLSR e do protocolo OLSR com as modificações propostas.

No Capítulo 4 ("Identificação de Nós Âncora") é introduzido o conceito de ligações estáveis, sendo também apresentado um algoritmo capaz de identificar este tipo de ligações. Posteriormente, expande-se o conceito de ligações estáveis para um nó, sendo apresentado um método capaz de agrupar os nós baseado na estabilidade das ligações que estes têm com os seus vizinhos. No final do capítulo, é avaliado o desempenho do algoritmo de agrupamento de nós.

O Capítulo 5 ("Encaminhamento com Optimização de Topologia") descreve o modo de funcionamento do protocolo OLSR, com especial atenção para a funcionalidade dos nós âncora (designados por nós MPR) característicos deste protocolo, e para o seu método de eleição. Na segunda parte, são apresentadas as modificações propostas ao método de eleição de nós âncora do protocolo OLSR.

No Capítulo 6 ("Análise de Desempenho") é avaliado o desempenho das modificações propostas ao protocolo OLSR, apresentadas no Capítulo 5. Começa-se por apresentar os detalhes das simulações realizadas, e no final do capítulo é elaborada uma análise comparativa entre o desempenho dos protocolos OLSR e OLSR modificado.

No Capítulo 7 ("Conclusões") é feita uma análise global do trabalho realizado, sendo também apresentadas algumas ideias para possíveis trabalhos futuros.

O Apêndice A ("Aplicação Auxiliar para a Ferramenta SUMO") apresenta uma aplicação desenvolvida em MATLAB durante a realização do projecto desta dissertação, que tem como função auxiliar o utilizador a gerar cenários de mobilidade, em formato de auto-estrada, através da ferramenta SUMO. A descrição da aplicação é acompanhada pela explicação da utilização da ferramenta SUMO.

Por fim, no Apêndice B ("*Script* de Simulação do *ns-2*") é descrita a estrutura dos *scripts* de simulação (do simulador de redes *ns-2*), utilizados para avaliar o desempenho dos protocolos OLSR e OLSR modificado, enquanto que no Apêndice C ("Publicações") são

apresentadas as publicações elaboradas ao longo desta dissertação.

Capítulo 2

Redes Veiculares

As redes veiculares são, hoje em dia, alvo de investigação tendo como objectivo a curto prazo, a melhoria da segurança nas estradas. Com base no trabalho já realizado, neste capítulo são apresentados os principais aspectos que caracterizam uma rede veicular.

O capítulo está dividido em três secções. Numa primeira secção é feita uma caracterização da rede veicular, com especial atenção para as particularidades que ela apresenta. Na segunda secção descreve-se o estado da arte relativamente a diversos desenvolvimentos realizados na camada de Controlo de Acesso ao Meio (*Medium Access Control*), enquanto que na terceira e última secção são referidos vários protocolos elaborados a nível de encaminhamento (*routing*).

2.1 Introdução

As redes veiculares, ou também conhecidas por *Vehicular Ad-hoc NETWORKS* (VANETs), *Inter-Vehicle Communications* (IVC) ou ainda *Vehicle-to-Vehicle* (V2V) *communications*, são actualmente uma tecnologia em ascensão, sendo alvo de inúmeros focos de investigação por todo o Mundo. Os grandes objectivos de estudo deste tipo de tecnologia são fornecer uma ligação entre veículos (nós móveis) e utilizadores fixos, como também, prover uma comunicação eficiente entre os veículos, de modo a permitir a utilização de Sistemas Inteligentes de Transportes.

Os Sistemas Inteligentes de Transporte, ou *Intelligent Transportation System* (ITS), são actualmente a mais importante aplicação de uma rede veicular, fornecendo serviços de segurança rodoviários [XSJC03] ou informações relativas a situações de tráfego intenso [WER⁺03]. Com o crescimento do uso da Internet, tornou-se também necessário prover aos ocupantes dos veículos o seu acesso [MWL03], disponibilizando serviço de *e-mail*, *download* de músicas, serviços de entretenimento a bordo, entre outros, representando também aplicações de elevada importância em redes veiculares.

As redes veiculares têm despertado grande interesse tanto a nível académico como a nível industrial. Ao longo dos anos tem-se notado um grande aumento do número de conferências que abordam esta temática, em conjunto com inúmeros estudos que têm sido feitos a nível académico. Por outro lado, grandes companhias automóveis, como a DaimlerChrysler, Audi, BMW, Fiat, Renault, Volkswagen, uniram-se com o intuito de criar um consórcio denominado *Car2Car Communication Consortium* (C2CCC) de modo a padronizar um sistema de comunicação *Car-to-Car* baseado em tecnologia *Wireless LAN* (WLAN) [CAR]. Também o *Institute of Electrical and Electronics Engineers* (IEEE) demonstrou o seu interesse em redes veiculares, criando um grupo de trabalho, 802.11p, com o objectivo de adaptar a rede sem fios, 802.11, de modo a esta poder suportar aplicações ITS.

As redes veiculares podem ser consideradas um caso especial de *ad-hoc* redes móveis, *Mobile Ad-hoc NETWORKs* (MANETs), e podem ser categorizadas consoante o tipo de ligações existentes. Assim, consideremos as três arquitecturas de redes veiculares:

- **Arquitectura WLAN ou celular:** Antenas fixas (futuramente designado de *gateway*), colocadas ao longo da estrada, funcionam como *access points* (pontos de acesso) fornecendo acesso à Internet. Não existe qualquer ligação directa entre veículos. Num cenário de auto-estrada, a implantação de *gateways* suficientes de modo a garantir cobertura necessária pode tornar-se bastante dispendiosa.
- **Arquitectura *ad-hoc*:** É considerada uma arquitectura *ad-hoc* quando existem apenas ligações entre nós móveis, sem existir necessidade de recorrer a infra-estruturas

fixas. Factores como a velocidade ou a densidade de nós podem pôr em causa o desempenho deste tipo de redes, como será explicado mais à frente nesta secção.

- **Arquitectura Híbrida:** Com o intuito de colmatar as falhas existentes nas duas arquitecturas anteriores, considerou-se uma outra arquitectura, denominada de arquitectura híbrida, também conhecida por *Wireless Mesh Network*, que é composta simultaneamente por redes *ad-hoc* e por redes WLAN.

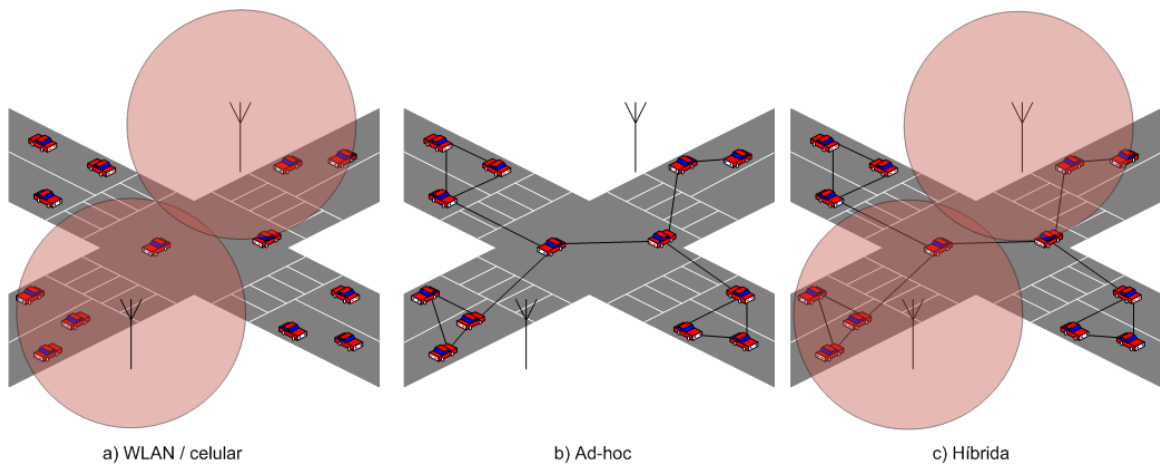


Figura 2.1: Arquitecturas de redes veiculares.

Todos os cenários de redes veiculares utilizados durante o projecto desta dissertação são baseados numa arquitectura *ad-hoc*.

Herdando das redes móveis todas as características que esta apresenta, tal como a não utilização de infra-estruturas fixas para reencaminhamento de informação, limitação de energia ou ainda a constante modificação de topologia, a rede veicular veio adicionar algumas características específicas:

- **Rápida mudança da topologia da rede:** A elevada velocidade dos nós faz com que a topologia da rede seja altamente dinâmica, impossibilitando assim o conhecimento da mesma a longo prazo.
- **Fragmentação da rede:** Adicionando o factor velocidade a situações de tráfego de baixa densidade, a distância entre veículos pode por vezes ser de quilómetros, indo

para além do alcance suportável pela ligação sem fios. O pouco tempo em que os nós tendem a ficar em contacto pode impossibilitar a criação de ligações e transferência de dados, dando origem a uma rede fragmentada [WFR04].

- **Energia suficiente:** Tendo em conta que numa rede veicular os nós são agora veículos ao invés de dispositivos portáteis, o problema de energia é corrigido pela possível utilização das baterias dos veículos.
- **Diversidade de ambientes comunicativos:** Podem ser definidos dois ambientes relativamente à aplicação de redes veiculares. Estas podem ser utilizadas num cenário de auto-estrada em que, na maioria dos casos, o seu ambiente não apresenta grandes obstáculos físicos. Ou então, podem ser utilizadas num cenário citadino, em que a existência de edifícios ou árvores fazem a separação das estradas, impossibilitando a criação de uma linha directa de comunicação.

2.2 Sub-camada MAC

O controlo de acesso ao meio (*Medium Access Control*) é uma sub-camada existente na camada lógica, segundo o modelo OSI (*Open Systems Interconnection Reference Model*). Tem como função definir um conjunto de regras e técnicas de acesso a um canal de comunicações. Sempre que existir uma colisão de dados no canal, este protocolo deverá indicar o procedimento a ser executado, assim como o tempo de espera de um nó antes de este poder retransmitir. O endereçamento é realizado pelo endereço MAC, também denominado de endereço físico¹, possibilitando o envio de pacotes para um destino, independentemente da rede a que este pertença.

As especificidades deste tipo de redes também representam vários obstáculos ao bom funcionamento dos protocolos MAC, as quais se fazem sentir nomeadamente no *throughput* (débito) da rede. As principais dificuldades são sumarizadas de seguida:

- Os *gateways* encontram-se na sua maioria distribuídos e nem sempre são alcançáveis pelos veículos, o que impossibilita uma ligação directa entre ambos. Durante este

¹Número identificativo de cada dispositivo de rede.

período, são necessárias ligações *ad-hoc* entre veículos, de modo a fornecer canais alternativos.

- A elevada velocidade dos veículos deteriora a qualidade das ligações, até mesmo quando se trata de ligações entre os veículos e o *gateway*.
- Diferentes veículos podem ter canais de comunicação com distintas condições, devido a *multi-path fading*² e a outras características respeitantes a canais sem fios (*path loss*³, *shadowing*⁴).

Nas sub-secções seguintes, são apresentados alguns exemplos de protocolos MAC utilizados em protocolos IVC, juntamente com os seus princípios básicos de funcionamento.

2.2.1 Métodos de Alocação Estática de Recursos

Relativamente aos métodos de controlo de acesso ao meio, uma das abordagens possíveis é o método de alocação estática de recursos. O conceito fundamental deste esquema reside no facto de ser estabelecido um canal de comunicação entre dois nós, mesmo antes de decidirem comunicar entre si.

Esta abordagem é a menos utilizada no desenvolvimento de protocolos MAC para redes veiculares, na medida em que a constante mudança topológica de uma rede veicular coloca a estabilidade e a durabilidade das ligações em causa. Como tal, e a título ilustrativo, é apresentado na sub-secção seguinte apenas um protocolo MAC baseado neste método.

ASDM

Tendo em conta as dificuldades impostas pela implementação de um protocolo MAC baseado na alocação estática de recursos, Blum et. al apresentaram em 2005 um protocolo denominado de *Adaptive Space Division Multiplexing* [BE05]. Este protocolo baseia-se no esquema de acesso ao meio *Space-Division Multiple Access* (SDMA), que consiste na mul-

²Soma dos vários sinais no receptor, tornando-o imperceptível.

³Enfraquecimento do sinal devido à distância percorrida.

⁴Enfraquecimento do sinal devido à existência de obstáculos.

tiplexagem espacial de canais de comunicação, consoante a localização geográfica dos nós.

O funcionamento do protocolo ASDM está dividido em três etapas:

- Divisão da estrada em várias células de forma a que na mesma célula só possa existir um veículo.
- Associação de cada célula a um *timeslot* (intervalo de tempo). Esta etapa, também chamada de função de mapeamento de *timeslots* (*Timeslot Mapping Function*), tem como principal objectivo distribuir os *timeslots* pelas células de forma mais equitativa.
- Criação de regras que permitem a associação de um veículo a um *timeslot*. O modo de funcionamento do SDMA permite que cada veículo possa transmitir no *timeslot* correspondente à célula que ele ocupa. No entanto, neste novo protocolo, um veículo pode não só transmitir no *timeslot* relativo à sua célula, mas também em todos os *timeslots* correspondentes às células vazias que se situam entre ele e o veículo que o precede.

No entanto, para que este protocolo funcione é necessário cumprir três requisitos: (1) todos os veículos têm de ser capazes de saber a sua posição e a do veículo que lhes precede (utilizando, por exemplo, sistemas de posicionamento), (2) todos os veículos têm de conhecer a função de mapeamento (3) e todos os veículos têm de ter acesso a um relógio comum (por exemplo, fornecido pelo mesmo sistema de posicionamento).

Este protocolo mostrou ser vantajoso relativamente a outros protocolos baseados em SDMA, no que diz respeito a uma melhor ocupação da largura de banda devido à sua remodelada função de mapeamento de *timeslots*.

2.2.2 Métodos de Alocação Dinâmica de Recursos

Tendo em conta as características particulares de redes veiculares, o método de alocação dinâmica de recursos é o mais apropriado para a realização de protocolos MAC. Nesta aproximação não existe um estabelecimento de ligação prévio entre nós, verificando-se

apenas a criação de canais de comunicação, e consequente ocupação do canal, quando existe a necessidade do envio de pacotes, o que garante um melhor aproveitamento da largura de banda. No entanto, esta aproximação apresenta um *throughput* variável e um maior atraso nas comunicações. Nas sub-seções seguintes são apresentados alguns exemplos de protocolos implementados utilizando esta técnica de controlo de acesso ao meio.

IEEE 802.11

A norma IEEE 802.11 propõe um protocolo MAC para redes sem fios (WLANS) e foi elaborada pelo grupo de trabalho IEEE 802. Existem diversas variantes desta norma fornecendo diferentes velocidades de transmissão, entre elas, o 802.11b que funciona na gama dos 2.4 GHz oferecendo taxas de transmissão na ordem dos 11 Mbps, e o 802.11a que funciona a 54 Mbps utilizando *Orthogonal Frequency-Division Multiplexing* (OFDM) na gama de 5 GHz.

O protocolo IEEE 802.11 suporta dois modos de funcionamento. O primeiro, denominado de *Distributed Coordination Function* (DCF), é um modo completamente distribuído, não utilizando nenhum tipo de controlo central, enquanto que o outro modo, chamado de *Point Coordination Function* (PCF), utiliza um nó base para controlar a actividade na sua célula. Tendo em conta que o cenário de mobilidade utilizado no projecto desta dissertação são redes veiculares com uma arquitectura *ad-hoc*, vai ser focado apenas o modo de funcionamento distribuído, DCF, do protocolo IEEE 802.11.

Este modo de funcionamento recorre a um método de acesso ao meio denominado de *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA), onde sempre que um nó deseje transmitir, e antes de o fazer, é obrigado a escutar o canal (*carrier sense*), de modo a perceber se este se encontra ocupado. O CSMA/CA apresenta dois modos de funcionamento, descrito nos parágrafos seguintes.

Ao escutar o canal, um nó consegue perceber o seu estado. Caso o canal esteja livre o nó transmite imediatamente o pacote completo, não voltando a escutar o canal até o envio

do pacote estar terminado. O pacote poderá não ser recebido com sucesso devido a interferências na recepção, geradas pelo problema da estação oculta. Na situação do canal se encontrar ocupado, o nó aguarda um período de tempo aleatório, utilizando o algoritmo de *binary exponential backoff*, voltando depois a tentar a transmissão.

Com o intuito de corrigir o problema da estação escondida, o CSMA/CA tem outro modo de funcionamento, baseado no protocolo *Multiple Access with Collision Avoidance for Wireless* (MACAW).

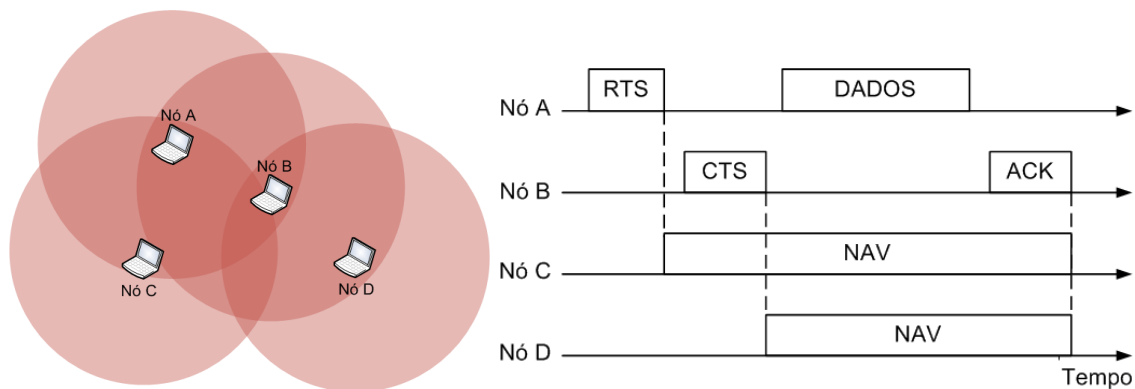


Figura 2.2: Funcionamento do protocolo CSMA/CA.

Considere-se o exemplo em que o nó A deseja enviar informação para B (ver Figura 2.2). O nó C encontra dentro do alcance do A, enquanto que D é um nó que está dentro do alcance de B, mas fora do alcance de A. No instante em que A decide comunicar com B, este envia um pacote denominado RTS (*Request To Send*) para B, de modo a pedir informar que deseja transmitir. Quando o nó B recebe o seu pedido, e estiver pronto para receber a transmissão, responde para o nó A com um pacote CTS (*Clear To Send*). Quando o pacote CTS chega ao nó A, este começa a transmitir e inicia um relógio de confirmação (*ACKnowledge timer*), que não deverá expirar antes de receber um pacote ACK de B com a confirmação de que a transmissão está completa. Caso o relógio de confirmação chegue ao fim, todo o processo é reiniciado.

Relativamente ao nó C, este deverá ter escutado o pacote RTS enviado por A, e assim sabe que vai existir troca de informação durante um certo período de tempo (informação

contida no pacote RTS), impedindo-o de fazer transmissões durante esse período. O nó D, não pôde escutar o pacote RTS, mas no entanto recebeu o pacote CTS, com o qual é também informado de que o canal estará ocupado. Estes impedimentos internos relativamente aos nós C e D são representados por uma ocupação virtual do canal também denominada de *Network Allocation Vector* (NAV).

O desempenho do protocolo MAC IEEE 802.11 é prejudicado pela topologia altamente dinâmica característica das redes veiculares, na medida em que os tempos de associação entre nós são relativamente elevados, principalmente se estivermos perante a utilização de aplicações ITS.

WAVE

Wireless Access in Vehicular Environments (WAVE), também conhecido por 802.11p, é um protocolo MAC ainda em desenvolvimento pelo grupo de trabalho IEEE 802.11, baseado no padrão IEEE 802.11, de modo a fornecer um sub-nível MAC e físico fiável (oferecendo uma latência baixa entre os 100 microssegundos e os 50 milissegundos, um alcance rádio até os 1000 metros, e um bom funcionamento do protocolo em veículos com velocidades máximas até 200 km/h) em cenários de redes veiculares, suportando aplicações ITS.

Utiliza o mesmo esquema de acesso ao meio do IEEE 802.11, o CSMA/CA, porém espera-se que estas aplicações operem na banda de frequência de 5.9 GHz nos Estados Unidos, e de 5.8 GHz na Europa e no Japão com uma taxa de transferência entre os 3 e os 27 Mbps.

V-PEACE

Em 2004, foi proposto um novo protocolo MAC denominado de *Vehicle Position Environment Acquisition and Communication Evolution* (V-PEACE) [NKMK04] baseado no protocolo CSMA/CA, mas com informação sobre localização. A Figura 2.3 esquematiza o funcionamento deste protocolo. A tecnologia *Time Division Multiplexing* (TDM) é utilizada para criar tantas tramas quanto as faixas que existam na estrada. Assim, cada veículo pode transmitir na trama TDM correspondente à sua faixa, no instante que de-

pende da posição do veículo na estrada em relação a um ponto de referência, pelo que é necessário que cada veículo esteja equipada com um sistema de posicionamento, como o *Global Positioning System* (GPS). Cada veículo tem direito a um pacote de transmissão com a duração correspondente ao comprimento do veículo. Assim, como teoricamente não é possível existirem dois veículos na mesma posição, não se verifica colisão de sinais e cada veículo fica a conhecer o tamanho dos veículos vizinhos.

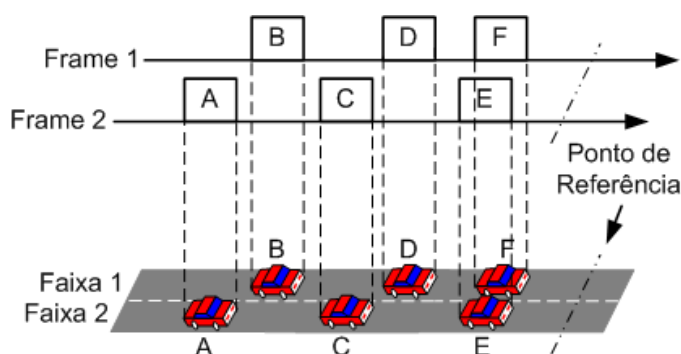


Figura 2.3: Esquema de funcionamento do protocolo V-PEACE.

Este protocolo, que necessita de um sistema de posicionamento com alta resolução, mostrou ter melhores resultados no que diz respeito à taxa de sucesso de entrega, relativamente aos protocolos CSMA e CSMA/CA, em cenários de pouca densidade e quando é utilizado um valor de *bit-rate* baixo (entre 1 e 5 Mbps).

WTRP

O *Wireless Token Ring Protocol* (WTRP) [LAP⁺01] é um protocolo MAC desenvolvido para redes veiculares aplicado a Sistemas Inteligentes de Transportes. É baseado no antigo protocolo *Token Ring* em que todos os participantes da rede formam um anel. Cada nó tem apenas ligação com dois outros nós, dentro desse anel circula um *token*⁵, que não é mais do que uma trama que serve para coordenar a comunicação na rede (ver Figura 2.4). Assim, sempre que um nó deseje comunicar, tem de esperar que o *token* chegue até si. Caso este chegue vazio poderá utilizá-lo para realizar uma transmissão, senão terá de aguardar que o *token* continue a circular no anel até chegar à sua vez de transmissão.

⁵Testemunho indicativo de permissão.

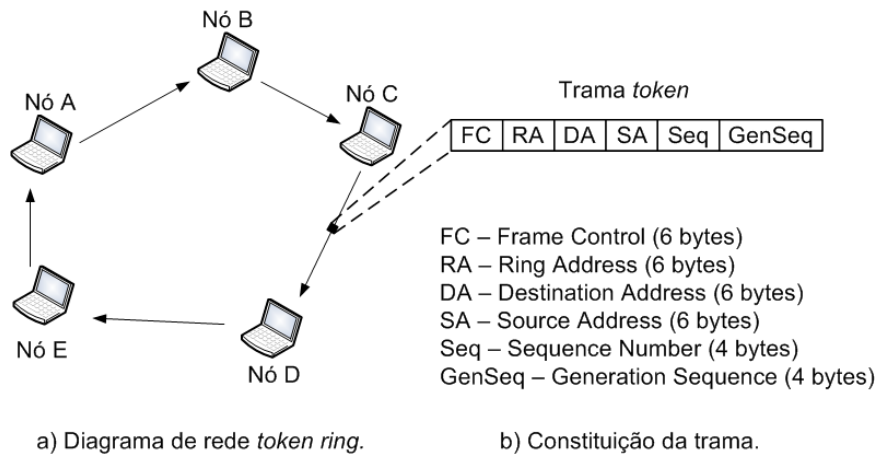


Figura 2.4: Funcionamento de transmissão do protocolo WTRP.

Este protocolo caracteriza-se pelo modo como recupera de várias falhas simultâneas, tais como a saída inesperada de algum nó da rede ou ainda a possível existência de mais do que um *token* na rede. O facto de só existir um *token* em cada anel evita que sucedam colisões, no entanto poderá aumentar o atraso *end-to-end* da rede, caso esta seja composta por muitos nós.

VC-MAC

Um dos mais recentes protocolos para redes veiculares, *VehiCular-MAC* (VC-MAC) [ZZJ07], desenhado principalmente para redes com arquitectura híbrida (descrito na Secção 2.1), é baseado nos benefícios de comunicação cooperativa⁶ juntamente com o conceito de reutilização espacial (aplicado a cenários de difusão).

A ideia principal deste protocolo, definido pelos autores como protocolo MAC, é a selecção de um conjunto de nós que apresentem ter canais de comunicação com melhores condições, que irão funcionar como *relays* (repetidores) de informação, o que adiciona a este protocolo uma componente de encaminhamento além da componente de controlo de acesso ao meio.

Este protocolo veio explorar de uma forma mais eficiente o conceito de reutilização espa-

⁶Quando um canal entre dois nós é inseguro, é escolhido um outro que tenha um canal em melhores condições para realizar encaminhamento, de modo a fornecer diversidade de rotas [NHH04].

cial, seleccionando um conjunto de *relays* ao invés de apenas um, como previamente se tinha implementado nos protocolos MAC cooperativos CoopMAC I, CoopMAC II [LTP05] e CMAC [NCG05].

Este protocolo encontra-se dividido em quatro estágios. Primeiramente, e durante um período de tempo denominado de *Broadcast of Gateway Period*, os *gateways* difundem informação para todos os veículos que se encontrem dentro do seu raio de alcance em modo *broadcast*, ou seja, sem qualquer tipo de compromisso - *handshake* ou mensagem de confirmação. De seguida, também durante um certo período de tempo (*Information Exchange Period* subdividido em *Relay Access Period* e em *Destination Access Period*), potenciais *relays* e potenciais destinatários respondem informando os seus vizinhos da sua existência, ficando assim a conhecer-se a topologia da rede. Após este período, surge um outro, *Relay Set Selection Period*, onde é feita a selecção do melhor conjunto de *relays*, segundo um modelo WIS - *Weighted Independent Set* (modelo de conjuntos independentemente ponderados). Finalmente, os nós seleccionados para o conjunto de *relays* fazem o encaminhamento dos pacotes, para os potenciais destinatários que se encontrem dentro dos seus raios de alcance.

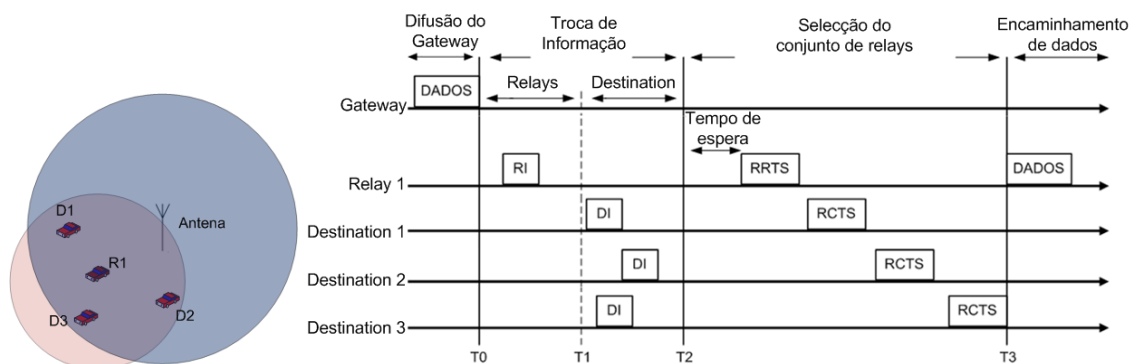


Figura 2.5: Troca de pacotes do protocolo VC-MAC.

2.3 Protocolos de Encaminhamento

Os protocolos de encaminhamento, como o nome indica, têm como objectivo calcular as rotas necessárias para encaminhar os pacotes com sucesso, desde a origem até ao destino.

Ao longo dos anos, realizaram-se diversos estudos com a finalidade de comparar o desempenho entre os vários protocolos existentes para redes móveis, aplicados a vários cenários de redes veiculares [JBW05], [JM96], [NAG04]. De modo a corrigir a sua degradação, alguns sofreram adaptações, enquanto que outros foram criados.

Nas sub-seções seguintes é feito um estudo sobre os principais tipos de encaminhamento a nível veicular, acompanhados por alguns exemplos.

2.3.1 Encaminhamento *ad-hoc*

As redes *ad-hoc* móveis tiveram a sua origem, em 1970, com o aparecimento da rede DARPA - *Defense Advanced Research Projects Agency*. Como estudado anteriormente, caracterizam-se por ser um tipo de redes não infra-estruturadas e sujeitas à mobilidade dos nós, e como tal, os protocolos de encaminhamento existentes para redes infra-estruturadas não apresentavam um desempenho aceitável neste novo tipo de redes. Assim sendo, foram desenvolvidos novos protocolos de encaminhamento para redes *ad-hoc* móveis, considerando as suas limitações: baixo consumo energético, baixa largura de banda e uma elevada taxa de erros.

Os protocolos de encaminhamento para redes *ad-hoc* móveis são tradicionalmente divididos em duas grandes categorias: reactivos e pro-activos. Os protocolos reactivos (*on-demand*), caracterizam-se pelo facto de nem sempre terem disponíveis na sua tabela de encaminhamento, as rotas para todos os nós da rede. Este tipo de protocolos foi desenhado para que uma rota só seja calculada apenas quando é necessária. Quando um nó necessita de uma rota para um certo nó na rede, inicia um processo de descoberta de rota, estando terminado quando a rota é calculada com sucesso ou, caso não exista rota disponível para o nó, depois de verificados todos os nós existentes na rede. A manutenção das rotas já determinadas vai sendo realizada enquanto o nó de destino permanecer alcançável, ou até já não ser mais necessária. Os protocolos de encaminhamento reactivos mais conhecidos são o *Ad hoc On Demand Distance Vector* AODV [PR99] e o *Dynamic Source Routing* (DSR) [JM96].

Os protocolos pro-activos (*table-driven*), ao contrário dos protocolos reactivos, caracterizam-se em ter o conhecimento das rotas para todos os nós existentes na rede, se tal for possível, a qualquer instante. Estes tipo de protocolos tem a vantagem de oferecer um atraso de envio inicial reduzido, pois a rota pode ser seleccionada da tabela de encaminhamento sempre que seja necessária. No entanto, a metodologia dos protocolos pro-activos obriga a existência de tráfego de controlo de topologia adicional, pois é necessário manter as rotas existentes na tabela de encaminhamento sempre actualizadas. Por exemplo, se estivermos na presença de uma rede móvel com uma densidade de nós bastante elevada, a probabilidade da quebra de ligações existentes é maior, originando assim mais tráfego de *broadcast*, e consequentemente uma maior ocupação da largura de banda para repor a rota. São exemplos de protocolos de encaminhamento pro-activos o *Destination Sequenced Distance Vector* (DSDV) [PB94] e o *Wireless Routing Protocol* (WRP) [MGLA96].

Mais recentemente surgiu outro tipo de encaminhamento para redes *ad-hoc*, baseado na optimização da topologia. O protocolo de encaminhamento com optimização de topologia mais utilizado em redes *ad-hoc* é o *Optimized Link State Routing* (OLSR). Na realização do projecto desta dissertação, foi adoptado o protocolo OLSR (com algumas modificações propostas) e, como tal, a descrição deste protocolo será retomada num capítulo futuro.

Alguns estudos, demostram que o desempenho dos protocolos AODV e DSR é bastante prejudicado pela frequente alteração topológica da rede, influenciando o *delay end-to-end*⁷ e a ocupação da largura de banda [Bou04]. Como tal, estes protocolos têm vindo a ser adaptados para redes veiculares.

PRAODV / PRAODV-M

Em 2004, Namboodiri et al. [NAG04] trabalharam sobre o protocolo AODV, oferecendo-lhe uma componente baseada na velocidade, na localização e na predição, resultando no *Preemptive AODV* (PRAODV) e no *Preemptive AODV-Maximum* (PRAODV-M). Estes novos protocolos diferem na medida em que o PRAODV estabelece uma ligação alternativa

⁷Tempo que decorre entre o envio da mensagem e a sua recepção no destino.

entre dois nós, antes de a existente expirar, ao contrário do AODV que só estabelece uma nova ligação quando é detectada a sua inexistência. Relativamente ao protocolo PRAODV-M, é escolhida a rota que se prevê ficar operacional durante mais tempo, contrariamente ao AODV onde é escolhida a rota mais curta. Estes protocolos revelaram algumas melhorias relativamente à taxa de sucesso de entrega de pacotes, mas devido à sua componente de estimação tornam-se pouco fiáveis.

2.3.2 Encaminhamento Baseado na Localização

É legítimo afirmar que numa rede veicular a movimentação dos veículos é sobretudo bidireccional, devido à disposição das estradas e ruas. Este facto pode ser aproveitado na criação de protocolos de encaminhamento baseados na localização, sendo no entanto necessário recorrer a mapas das estradas, informação geográfica ou até a sistemas de posicionamento incorporados no veículo.

GPSR

Um dos primeiros protocolos baseados na localização é o *Greedy Perimeter Stateless Routing* (GPSR) [KK00]. Este protocolo baseia-se apenas na informação geográfica sobre os vizinhos de modo a realizar encaminhamento *greedy*⁸. Exemplificando, como se pode observar na Figura 2.6, quando o nó S recebe um pacote que precisa de enviar para o nó R, não existindo comunicação directa entre ambos, o nó S vai recorrer ao nó que estiver mais próximo geograficamente do nó de destino, e assim sucessivamente. Um dos pontos fortes deste algoritmo é o de manter informação apenas sobre a topologia local, permitindo uma melhor escalabilidade e uma redução no tempo utilizado no cálculo de novas rotas. No entanto, o desempenho deste algoritmo tem o seu expoente máximo em cenários sem obstáculos físicos, de preferência com os nós uniformemente distribuídos.

De modo a corrigir a limitação do algoritmo estudado anteriormente, foram propostos dois protocolos de encaminhamento baseados na posição/localização, focados em ambientes citadinos: o *Global State Routing* (GSR) [CG98] e o *Anchor-based Street and Traffic Aware Routing* (A-STAR) [LLL⁺04].

⁸Também conhecido como encaminhamento ávido.

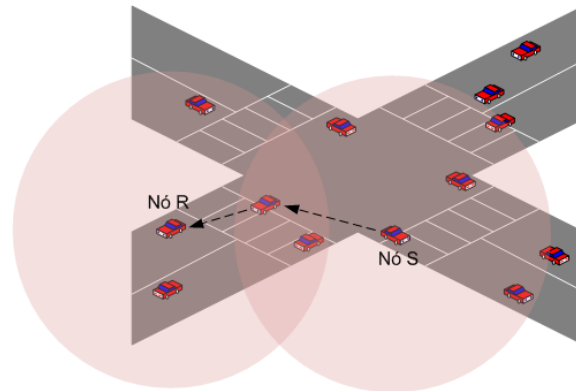


Figura 2.6: Exemplo de encaminhamento *greedy*.

GSR

O GSR é um protocolo baseado na localização, apoiado por informação topológica, que utiliza *link-state* [MRR80]. Cada nó mantém uma tabela de conectividade, contendo todas as ligações existentes entre os diversos nós da rede, o que otimiza as decisões a nível de encaminhamento local. Depois de descoberta a localização do nó, com recurso a inundação, é utilizado um mapa digital das estradas de modo a calcular o conjunto de troços que o pacote terá de seguir. Este cálculo é efectuado recorrendo ao algoritmo de *Dijkstra*. Estudos realizados comprovam que este protocolo apresenta melhorias acentuadas relativamente ao GPSR em matéria de atraso *end-to-end* comparativamente com o DSR [FMH⁺03], assim como uma melhor taxa de sucesso de entrega e menor ocupação da largura de banda face ao AODV [LW07].

A-STAR

O protocolo A-STAR baseia-se no GSR e no GPSR relativamente à utilização de mapas das estradas para calcular o conjunto de troços através do qual o pacote deverá passar para atingir o seu destino. No entanto, difere em dois aspectos importantes: o A-STAR incorpora um sistema de sensibilização de tráfego (*traffic awareness*) recorrendo a mapas das estradas ordenados por utilização, de modo a poder definir as suas rotas pelas estradas com maior conectividade, garantindo assim uma maior probabilidade no sucesso de entrega; por outro lado, este protocolo emprega uma nova estratégia de descoberta de recursos locais limitando-a a um certo valor. Graças à sensibilização de tráfego apli-

cado neste protocolo, este apresenta um melhor desempenho, nomeadamente mais 40% de sucesso de entrega de pacotes, comparado com o GSR.

GeOpps

Leontiadis et. al em 2007 propuseram um novo protocolo de encaminhamento intitulado de *Geographical Opportunistic routing for vehicular networks* (GeOpps) [LM07]. Este algoritmo, que assume que todas os veículos estão equipados com sistemas de posicionamento, explora as informações geográficas por estes obtidas, de modo a encaminhar um pacote para um nó que supostamente está em melhores condições (posicionalmente) de o poder entregar ao seu destino final. Resultados mostram que este protocolo tem um comportamento melhor do que o GPSR [KK00], estudado anteriormente, colmatando a falha de que pacotes enviados por nós que não fazem parte da mesma área raramente sejam entregues.

2.3.3 Encaminhamento Baseado em *Clusters*

Os algoritmos de encaminhamento baseados em *Clusters* (grupos) podem ser definidos como redes virtuais criadas através dos nós da rede física. O *cluster* é formado por um conjunto de veículos interligados entre si de forma lógica, conforme é ilustrado na Figura 2.7. Um *cluster* composto por nós móveis apresenta grandes diferenças relativamente a um *cluster* composto por nós fixos: um *cluster* móvel tem tendência a alterar rapidamente a sua composição devido ao movimento dos nós, e a sua principal função é apoiar a computação e actualização de rotas, ao invés de encaminhar pacotes.

Cada *cluster* pode conter um nó denominado de *cluster-head* (líder de grupo), que é responsável pela coordenação *inter* e *intra-cluster* relativamente a funções de gestão da rede. Os nós que compõem um *cluster* comunicam entre si directamente, enquanto que a comunicação entre *clusters* é feita pelos *cluster-heads*. A criação destes *clusters* é essencial para a escalabilidade dos protocolos de encaminhamento, e está na sua estabilidade a chave para o desempenho dos algoritmos baseados em *clusters*.

Devido ao curto tempo de vida de um *cluster*, os protocolos baseados em *clusters* dese-

nhados para redes móveis, tais como o *Adaptive Clustering* [LG97] e uma adaptação do *Minimum Connected Dominating Set* (MCDS) [DB97], tornam-se obsoletos quando aplicados a redes veiculares.

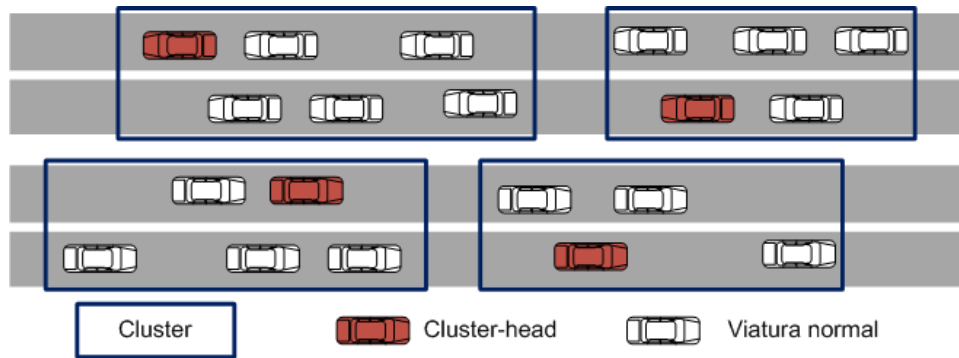


Figura 2.7: Formação de *clusters* numa rede veicular.

COIN

Em 2003 Blum et. al desenhou um protocolo de encaminhamento baseado em *clusters* denominado de *Clustering for Open IVC Networks* (COIN) [BEH03]. Este protocolo baseia-se no *Adaptive Clustering* com a utilização do *Minimum Connected Dominating Set* (conjunto de nós dominantes e conectados - CNDC), mantendo a base do método de eleição do líder de grupo e utilizando a informação sobre mobilidade para a criação dos *clusters*. No entanto, as intenções do condutor do veículo em conjunto com a dinâmica veicular são adicionadas ao algoritmo de criação de *clusters*. Estas alterações contemplam a natural variação de distâncias entre veículos, e resultados [BEH03] demonstram que as optimizações realizadas melhoram o desempenho do algoritmo, já que identificam um aumento de 192% no tempo médio de vida de um *cluster* e uma redução de 42% no número de alterações de membros de um *clusters*.

CBLR

Em 2004 Santos et. al [SEE04] apresentou um novo algoritmo de encaminhamento baseado em *clusters*, juntamente com informação sobre localização, de nome *Cluster-Based Location Routing algorithm* (CBLR). A metodologia deste algoritmo é a seguinte:

1. Sempre que um nó deseja enviar um pacote verifica se o nó de destino já se encontra na sua tabela de encaminhamento.
2. Caso a sua localização seja conhecida, o pacote é enviado. Caso contrário o pacote é guardado em memória, inicia-se um relógio e são enviados pacotes *Location REQuest* (LREQ) por *broadcast*.
3. Cada *cluster-head* ao receber o pacote LREQ verifica se o destino pertence ao seu *cluster*.
4. O *cluster-head* que contenha o destino no seu *cluster*, envia um *Location REPLY* (LREP) para o nó de origem, utilizando encaminhamento geográfico, pois cada *cluster-head* conhece a posição do nó de origem e o mais próximo *cluster-head*. Alternativamente, se um *cluster-head* não contiver o nó de destino dentro do seu *cluster*, o pacote LREQ é reencaminhado para os *cluster-head* adjacentes para prosseguir a busca.
5. Assim que o nó de origem receber a localização do nó de destino, o pacote é encaminhado para o *cluster-head* mais próximo.

Este algoritmo, comparativamente com outros não baseados em localização, tais como o AODV e o DSR, demonstrou um desempenho superior relativamente ao atraso *end-to-end* e à taxa de sucesso de entrega, à custa da informação sobre a localização.

O desempenho dos algoritmos de encaminhamento baseados em *clusters* serão sempre postos em causa devido à elevada dinâmica da topologia da rede, o que obrigará a um aumento de tráfego *broadcast* de forma a criar e a manter os vários *clusters*.

2.3.4 Encaminhamento por *Broadcast*

O mecanismo de transmissão *broadcast* (difusão) consiste em difundir informação por todos os nós que façam parte de uma rede. Muitos algoritmos de encaminhamento *unicast*, na sua fase de descoberta de recursos, utilizam este tipo de difusão de modo a encontrar o melhor caminho para um certo destino. Em VANETS, este tipo de difusão é muitas

vezes usado para partilha de informação de tráfego, condição das estradas, condições climáticas, entre outros.

A forma mais simples de implementar um serviço *broadcast* é recorrendo à técnica de inundação pura: cada nó reencaminha uma única vez a mensagem para todos os seus vizinhos, excepto para o nó de onde recebeu a mensagem. Esta técnica garante que todos os nós pertencentes à rede recebem a mensagem. Contudo, não deve ser aplicada em redes consideravelmente grandes, pois poderá originar um efeito de tempestade de broadcast (*broadcast storm*⁹), aumentando assim a probabilidade de colisões de pacotes e de ocupação de largura de banda, comprometendo o seu desempenho.

BROADCOMM

Em 2005, Durresi et. al desenvolveram um protocolo de emergências de encaminhamento baseado em *broadcast* recorrendo a informação geográfica, especialmente utilizado em auto-estradas. Este algoritmo, denominado de BROADCOMM (*BROADcast COMMunications*) [DDB05], apresenta semelhanças relativamente a outros algoritmo de encaminhamento baseados em *clusters* na medida em que a auto-estrada é dividida em células. No entanto, os *cluster-heads*, aqui chamados de *cell reflectors* são os nós que se encontrarem geometricamente no centro das células (ver Figura 2.8). A função dos *cell reflectors* é a de difundir as informações de emergência entre as suas células e difundi-las para os restantes *cell reflectors*. Este protocolo é bastante simples mas apenas funciona em cenários de auto-estradas.

2.3.5 Encaminhamento *Geocast*

O protocolo de encaminhamento *Geocast* (*Geocast Routing*) é também conhecido por encaminhamento multidifuso baseado na posição/localização (*Location-Based Multicast Routing*). O grande objectivo deste tipo de encaminhamento é o de entregar um pacote apenas aos nós que pertençam a uma certa região, também denominada de *Zone of Relevance* (ZOR). Uma implementação deste tipo de protocolo é a integração de um serviço

⁹Excesso de informação *broadcast* numa rede [NTCS99].

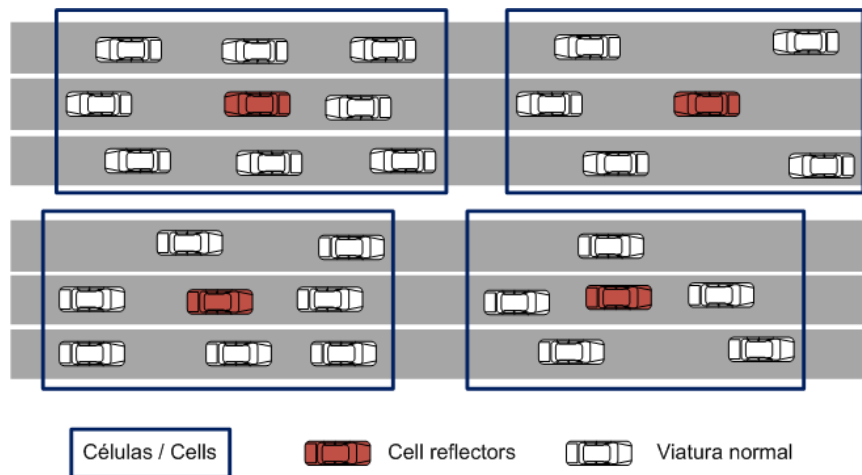


Figura 2.8: Constituição de células utilizado pelo BROADCAST.

de multidifusão em paralelo com o agrupamento dos nós conforme a sua localização geográfica, formando assim as várias ZORs.

Tomemos como exemplo um cenário de uma auto-estrada em que dois veículos do mesmo sentido embatem, causando um corte de todas as faixas do mesmo sentido. Utilizando este tipo de protocolo, só seriam avisados do acidente os veículos que supostamente iriam passar pela zona do acidente no mesmo sentido, ou seja, todas os veículos que circulavam atrás dos veículos acidentados, e que constituem assim a ZOR. Assim, qualquer veículo localizado no sentido contrário, que não se encontra dentro da ZOR, não tomaria conhecimento do acidente, de modo a evitar propagações de mensagens desnecessárias e consequentemente uma poupança na ocupação da largura de banda. Este exemplo encontra-se ilustrado na Figura 2.9.

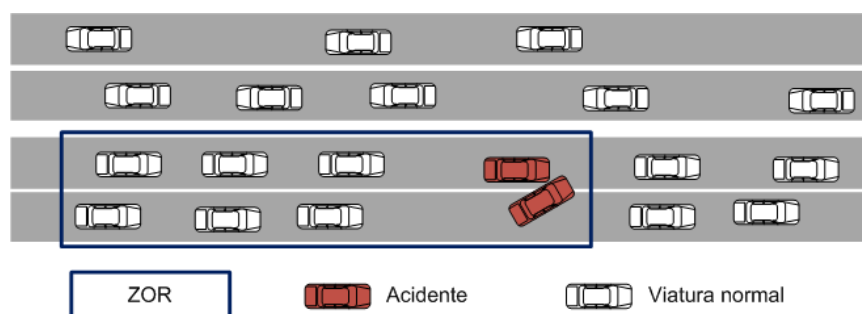


Figura 2.9: Constituição da ZOR em protocolos de encaminhamento *geocast*.

Message Dissemination Process

Em 2000, Linda Briesemeister et. al [BSH00] propuseram um protocolo de encaminhamento *geocast* com o objectivo de evitar a colisão de pacotes e de reduzir o número de retransmissões, referido pelos autores como *Message Dissemination Proccess*. Quando um nó recebe um pacote, não o reencaminha imediatamente esperando um certo período de tempo de modo a poder tomar uma decisão sobre a retransmissão. O período de tempo de espera depende da distância ao nó que lhe enviou o pacote: quanto maior for essa distância, mais curto é o tempo de espera. Quando o período de tempo expira, o pacote só é retransmitido se a mensagem não tiver sido recebida novamente. Esta optimização relativamente ao conceito fundamental de *broadcast* faz com que seja menos provável existirem *broadcast storms* e a disseminação de pacotes seja mais eficiente.

DRG / ROVER

Mais recentemente, em 2008, Kihl et. al implementaram dois algoritmos de encaminhamento *geocast*, o *Distributed Robust Geocast* (DRG) e o *RObust VEhicular Routing* (ROVER) [KSJ08]. O DRG é um protocolo adaptável à frequente mudança da topologia, que fornece um sistema de encaminhamento rápido e fiável que minimiza a carga na rede focalizado para grandes cenários. Por outro lado, o protocolo ROVER oferece uma difusão *multicast* geográfica fiável, baseada num processo de descoberta de rotas reactivo dentro da sua ZOR, inspirado no AODV, por forma a permitir a utilização de aplicações de Internet em conjunto com um protocolo de transporte fiável.

2.3.6 Comparação Entre Protocolos de Encaminhamento

Na Tabela 2.1 é apresentado um resumo comparativo dos protocolos de encaminhamento estudados anteriormente. A melhor estratégia para desenvolver um protocolo de encaminhamento aplicado a redes veiculares ainda não foi encontrada, existindo autores que consideram um encaminhamento baseado em *clusters* mais rentável do que um protocolo que se baseie fundamentalmente na difusão "exagerada" de mensagens (encaminhamento por *broadcast*). O cenário de aplicabilidade do protocolo faz também aumentar o número de propostas existentes: se existem algumas que têm um desempenho bastante favorável

em cenários citadinos, este decresce em cenários de alta mobilidade, e vice-versa.

Tabela 2.1: Algoritmos de encaminhamento aplicados a redes veiculares.

Protocolo de Encaminhamento	Tipo de Encaminhamento	Informação sobre Posição (Modo de uso)	Estrutura Hierárquica	Cenário de Mobilidade
AODV	<i>Unicast</i>	Não	Não	—
DSR	<i>Unicast</i>	Não	Não	—
OLSR	<i>Unicast</i>	Não	Não	—
PRAODV/-M	<i>Unicast</i>	Seleção de Rotas (mais predição tempo de vida)	Não	Auto-estrada (simples)
GPSR	<i>Unicast</i>	Encaminhamento de Pacotes	Não	—
GSR	<i>Unicast</i>	Encaminhamento de Pacotes (mais informação geográfica)	Não	Citadino (real)
A-STAR	<i>Unicast</i>	Encaminhamento de Pacotes (mais informação geográfica)	Não	Citadino (grelha)
GeOpps	<i>Unicast</i>	Encaminhamento de Pacotes (mais informação geográfica)	Não	Citadino (real)
COIN	<i>Unicast</i>	Formação de <i>Clusters</i>	Sim	Auto-estrada (real)
CBLR	<i>Unicast</i>	Encaminhamento de Pacotes (mais predição posicional)	Sim	Circuito circular e quadrangular
Flooding	<i>Broadcast</i>	Não	Não	—
BROADCASTMM	<i>Broadcast</i>	Formação de Células	Sim	Auto-estrada (simples)
Msg. Diss. Proc.	<i>Geocast</i>	Encaminhamento de Pacotes	Não	Auto-estrada (simples)
DRG	<i>Geocast</i>	Encaminhamento de pacotes (mais informação geográfica)	Não	Auto-estrada (simples)
ROVER	<i>Geocast</i>	Encaminhamento de Pacotes (mais informação geográfica)	Sim	Auto-estrada (simples)

Durante este capítulo introduziu-se o conceito de redes. Foram apresentadas as suas características e fundamentalmente, as principais exigências para o desenvolvimento de protocolos viáveis, tanto ao sub-nível MAC como ao nível de encaminhamento. Posto isto, descreveram-se alguns protocolos de sub-nível MAC já existentes e a sua tentativa de adaptação em cenários de redes veiculares, assim como protocolos mais recentes, especificamente desenvolvidos para este tipos de redes. Relativamente aos protocolos de encaminhamento, e sendo este nível o principal alvo de estudo desta dissertação, realizou-se um estudo mais extenso dos protocolos existentes apontando as suas vantagens e desvantagens, estando assim lançado o mote para a proposta que será apresentada nos capítulos que se seguem.

Capítulo 3

Modelos de Mobilidade

Uma das etapas mais importantes no desenvolvimento de um protocolo, com vista a ser utilizado em redes veiculares ou em qualquer outro tipo de redes, é o seu teste/validação. Em redes veiculares, para que o desempenho real de um protocolo seja satisfatório, é necessário que o modelo de mobilidade utilizado no cenário de teste reproduza de forma mais realista possível o meio onde este vai ser utilizado.

Este capítulo encontra-se dividido em três secções. Na primeira secção é feita uma revisão dos modelos de mobilidade existentes. Na segunda, são apresentadas algumas aplicações geradoras de modelos de mobilidade, com especial atenção para a aplicação utilizada durante o projecto desta dissertação, que gerou os cenários testados. Por último, na terceira secção são apresentados os cenários de simulação implementados e utilizados durante a realização do projecto desta dissertação.

3.1 Introdução

Os modelos de mobilidade desempenham um papel bastante importante no desenvolvimento de protocolos ou aplicações a serem utilizados em redes veiculares. Como se pode entender, é bastante complicado logisticamente e por vezes financeiramente, testar e validar os protocolos implementados em ambientes reais, o que relega para a simulação a avaliação do seu desempenho.

No desenvolvimento de qualquer aplicação, seja ela destinada à utilização em redes veiculares ou a qualquer outro tema, existe sempre a necessidade de fazer aproximar o mais possível o cenário de teste e de validação ao ambiente real correspondente. No que diz respeito às redes veiculares, e aos modelos de mobilidade utilizados, a sua principal dificuldade está em fazer com que os modelos reproduzam o máximo possível o comportamento do tráfego veicular e do ambiente circundante. Considera-se que um modelo de mobilidade realístico deverá contemplar os seguintes aspectos [HFB08]:

- **Mapas topológicos realistas:** os modelos de mobilidade deverão conter cenários o mais realistas possível, contendo ruas com diferentes densidades, diversidade no número de faixas e nas velocidades máximas associadas.
- **Acelerações e travagens suaves:** tendo em conta que, em situações normais, os veículos não realizam travagens nem acelerações bruscas, os modelos de aceleração e de travagem deverão ser contemplados no modelo de mobilidade.
- **Existência de obstáculos:** tanto a nível de mobilidade como a nível de comunicação, a presença de obstáculos existe e deverá ser considerada.
- **Pontos de atracção:** nenhum condutor tem um ponto de origem e de chegada aleatórios. Em muitos casos, o destino final de vários condutores é comum, criando situações de engarrafamento. Assim, pode-se considerar que um condutor se move a partir de um ponto de repulsão para um ponto de atracção, utilizando um percurso pré-definido.
- **Tempo de simulação:** o volume de tráfego não é uniforme ao longo do dia, existindo picos de aumento de tráfego em alturas conhecidas como *horas de ponta* ou aquando da existência de eventos especiais.
- **Distribuição de veículos não aleatória:** como se pode observar no dia a dia, os carros não podem ser distribuídos de forma uniforme no cenário de simulação, pois existem sempre focos de atracção onde a sua densidade é maior do que em outros pontos, tais como centros comerciais, zonas de escritórios e zonas residenciais.
- **Comportamentos de condução inteligentes:** os condutores reagem de acordo

com o ambiente, não apenas no que diz respeito a obstáculos físicos, mas também relativamente a situações de tráfego ou a existência de peões no meio da estrada, o que os faz mudar de percurso.

Os primeiros modelos de mobilidade a aparecer foram designados de *Random Node Movement*. Estes modelos, como o nome indica, são principalmente caracterizados pela sua aleatoriedade e, como tal, não espelham de forma alguma as características de uma rede veicular. No entanto, devido à sua simplicidade de parametrização, são ainda bastante utilizados em testes simples. Existem vários tipos de modelos baseados em movimento aleatório [CBD02]:

- ***Random Walk Mobility Model:*** este modelo é caracterizado por velocidades e direcções aleatórias.
- ***Random Waypoint Mobility Model:*** baseia-se no modo de funcionamento do modelo anterior, no entanto, quando o nó atinge o destino executa um tempo de pausa, recomeçando novamente o andamento.
- ***Random Direction Mobility Model:*** neste modelo, a velocidade e a direcção do nó são também escolhidas de forma aleatória, contudo, o nó só pára quando atingir o limite da área de simulação.

Com a necessidade de aproximar o comportamento dos modelos de mobilidade às características de uma rede veicular, alguns trabalhos utilizam modelos baseados em dados reais, os quais se denominam de *Real-World Trace Models*. Estes dados são extraídos de sistemas que contêm informação sobre o comportamento dos veículos numa estrada ou numa cidade real, como por exemplo, dos registos de informação do GPS. Esta é a informação mais realista que se pode utilizar, contudo estes modelos apresentam alguns inconvenientes tais como o elevado tempo de processamento das simulações, não podendo ainda ser parametrizáveis dado que representam uma situação real.

De modo a corrigir as limitações apresentadas pelos tipos de modelos anteriores, foi definido um novo tipo que, continuando a representar as características de uma rede veicular, oferece um maior grau de liberdade, permitindo a parametrização dos várias com-

ponentes do modelo, de modo a estudar a sua influência nos resultados das simulações. Actualmente, este tipo de modelo de mobilidade, designado de *Artificial Mobility Traces*, pode ser dividido em duas categorias, dependendo do nível de detalhe utilizado na criação do modelo [FHFB07]:

- **Traffic Stream Models:** este tipo de modelo é caracterizado como modelo macroscópico pois baseia o tráfego veicular em três variáveis fundamentais: a velocidade (*km/hora*), a densidade (*veículos/km*) e o fluxo de tráfego (*veículos/hora*). Estes modelos geram um fluxo médio de veículos baseado numa determinada distribuição (normalmente exponencial), estando normalmente associados a estudos teóricos baseados em teoria de fluxo. Como tal, raramente são utilizados em simulações pois, apesar de espelharem melhor a realidade das redes veiculares do que os modelos aleatórios, não representam o comportamento individual do condutor.
- **Car-following Models:** é um exemplo de um modelo microscópico na medida em que o comportamento de um veículo está directamente relacionado com o comportamento do veículo que lhe sucede. Em 1998, Krauß desenvolveu um modelo de *car-following* para uma estrada recta com dois sentidos, e uma faixa por sentido [Kra98]. Este modelo, é ainda a base de muitas aplicações de geração de modelos de mobilidade do tipo *car-following*, como por exemplo o SUMO (que será descrito mais à frente neste capítulo). O modelo baseia-se em quatro variáveis de entrada (a que representa taxa de aceleração, b a taxa de desaceleração, v_{max} indica a velocidade máxima e η que introduz aleatoriedade no modelo) e no seguinte conjunto de equações:

$$v_i^s(t + \Delta t) = v_{i+1}(t) + \frac{\Delta x_i(t) - v_{i+1}(t)\tau}{(v_i(t) + v_{i+1}(t))/2b + \tau} \quad (3.1)$$

$$v_i^d(t + \Delta t) = \min[v_{max}, v_i(t) + a\Delta t, v_i^s(t + \Delta t)] \quad (3.2)$$

$$v_i(t + \Delta t) = \max[0, v_i^d(t + \Delta t) - \epsilon a \Delta t \eta] \quad (3.3)$$

A Equação (3.1) calcula a velocidade necessária para que o veículo i mantenha uma distância de segurança relativamente ao veículo da frente. O tempo de reacção do condutor é representado por τ . A Equação (3.2) determina a nova velocidade

do veículo i escolhendo a menor velocidade entre a velocidade máxima, a velocidade actualizada dada a aceleração definida e a velocidade máxima que garante a distância de segurança face ao veículo que o sucede. Já a Equação (3.3) define a velocidade final do veículo depois de adicionada alguma aleatoriedade, dada por η , que caracteriza a imperfeição do condutor em impor ao veículo a velocidade $v_i^d(t + \Delta t)$ (qualquer erro ou atraso de reacção do condutor resultará numa velocidade inferior).

Os cenários de mobilidade utilizados no projecto desta dissertação foram obtidos recorrendo à ferramenta SUMO, e como tal caracterizam-se por serem modelos *car-following*.

Recentemente, tem vindo a ser desenvolvido um novo tipo de modelo de mobilidade, com participação em simultâneo dos simuladores de protocolos da rede e dos simuladores de tráfego, denominado de *Bidirectionally Coupled Simulators*. Este conceito baseia-se na ideia de partilhar, enquanto a simulação decorre, informação entre o simulador de protocolos da rede e o simulador de tráfego de forma a fornecer algum *feedback* ao condutor para que este possa tomar decisões durante a simulação. Nestas simulações, os dois simuladores partilham algumas informações, tal como a posição e a velocidade dos veículos em intervalos regulares, enquanto que outras, tais como as rotas planeadas, são localmente guardadas ou no simulador de redes ou no simulador de tráfego. Contudo, os resultados da simulação de tráfego não podem ser reutilizáveis na forma de ficheiros de *trace*, pois a mobilidade dos veículos é processada no momento. Assim, o modo de funcionamento deste tipo de modelo divide-se em duas fases:

- Enquanto a simulação decorre no simulador de protocolos da rede, são enviadas, para a simulação de tráfego, novos parâmetros por forma a alterar o comportamento do condutor ou os atributos da estrada. Nesta etapa, o tempo de simulação apenas avança no simulador de protocolos da rede.
- Baseado nos novos parâmetros enviados anteriormente pelo simulador de protocolos da rede, a simulação de tráfego processa os futuros movimentos dos veículos e envia-os de volta para o simulador de protocolos da rede. O tempo de simulação apenas avança no simulador de tráfego, e estas duas etapas repetem-se, em ciclo, até o tempo de simulação terminar.

Actualmente, já existem algumas aplicações que providenciam este tipo de simulação, nomeadamente: o *Traffic and Network Simulation environment* (TraNS) [PRL⁺07] que liga o SUMO com o simulador de redes ns-2 [Inf07]; e o *Vehicles in network simulation* (Veins) [SYGD08], que combina o SUMO neste caso com o simulador de redes OMNeT++.

Pode-se então concluir que desde o início do estudo das redes veiculares, os modelos de mobilidade utilizados no testes e validações de protocolos e aplicações têm sofrido profundas alterações, e diversos esforços têm sido levados a cabo de modo a que os cenários de mobilidade sejam cada vez mais realistas. Para finalizar, é apresentado na Tabela 3.1 um resumo dos vários modelos de mobilidade estudados.

Tabela 3.1: Modelos de mobilidade.

Modelo de Mobilidade	Aplicações Preparadas	Vantagens	Desvantagens
<i>Random Node Movement</i>	Qualquer Aplicação	+ Simplicidade	- Impreciso e Irreal
<i>Real World Traces</i>	JiST/SWANS, OPNET, GloMoSim, Qualnet, OMNeT++/INETm, ns-2	+ O Mais Realista + <i>Traces</i> Reutilizáveis	- Não Parametrizável - Dispendioso - Moroso
<i>Artificial Mobility Traces</i>	JiST/SWANS, OPNET, GloMoSim, Qualnet, OMNeT++/INETm, ns-2	+ Realista + Parametrizável + <i>Traces</i> Reutilizáveis	- Sem <i>feedback</i> no condutor
<i>Bidirectionnally Coupled Simulators</i>	Em desenvolvimento para OMNeT++/INETm, ns-2, Shawn, JiST/SWANS	+ Realista + Parametrizável + <i>Feedback</i> no condutor	- <i>Traces</i> não reutilizáveis

3.2 Aplicações Geradoras de Cenários de Mobilidade

Com o avanço do desenvolvimento de protocolos e aplicações para redes veiculares, e o consequente aumento da utilização de cenários de mobilidade para teste/validação das aplicações, novas ferramentas *open-source* (de código aberto) de geração de cenários de mobilidade têm surgido, produzindo assim os ficheiros de *trace* (registo) necessários para os simuladores de redes. Durante esta secção vão ser estudadas algumas aplicações existentes, com especial atenção para a aplicação utilizada no projecto desta dissertação.

GEMM

Em 2004, Feeley et. al implementaram uma ferramenta denominada de GEMM [FHR04], baseada sobretudo nas características da mobilidade humana. Como tal, foi uma das primeiras ferramentas a introduzir os seguintes conceitos: pontos de atracção, referidos na secção anterior; actividade, que consiste no deslocamento até um certo ponto de atracção, e lá permanecer durante um certo período de tempo e o conceito de *papel* que caracteriza várias tendências de mobilidade (características de várias classes de pessoas). Estes novos conceitos pretendiam simular e modelar padrões de mobilidade representativos de situações reais, no entanto, não passam de simples modelos *Random Waypoint Models* entre pontos de atracção.

MOVE

A aplicação *Mobility Model Generator for Vehicular Networks* (MOVE) [KML07] é escrita em Java, e baseia-se no simulador de micro mobilidade SUMO. Como tal é uma aplicação que suporta micro mobilidade, onde é possível importar mapas topológicos da base de dados TIGER [Bur], assim como produzir manualmente mapas pseudo aleatórios. Os cenários de mobilidade gerados por esta aplicação podem ser utilizados tanto no simulador de redes ns-2 [Inf07] como no Qualnet [Net].

STRAW

Choffnes et. al desenvolveram, em 2005, uma aplicação geradora de cenários de mobilidade denominada de *STreet RAndom Waypoint* (STRAW) [CB05]. Esta é baseada noutra ferramenta denominada *Scalable Wireless Ad Hoc Network Simulator* (SWANS) [Bar04]. Do ponto de vista de mobilidade veicular, e à semelhança da aplicação anterior, os mapas topológicos podem também ser importados da TIGER, ou então elaborados manualmente com suporte para micro mobilidade. Além disso, o STRAW é das poucas aplicações capazes de realizar um funcionamento complexo de cruzamentos, com possível recurso a sinais luminosos e a sinais de trânsito. Esta aplicação é prejudicada pelo facto da plataforma SWANS não ser das mais divulgadas.

VanetMobiSim

A aplicação *VanetMobiSim* [HFBF06], baseada na antiga *CanuMobiSim* [Stu], foi desenvolvida de modo a corrigir as limitações da sua antecessora, fornecendo algum realismo relativamente à mobilidade veicular. Apesar de fornecer uma arquitectura de mobilidade eficiente, a aplicação sofre de falta de detalhe em alguns cenários específicos, devido à sua natureza de uso geral. Esta aplicação suporta macro e micro mobilidade, existindo também a possibilidade de se extraírem mapas topológicos do TIGER, e revela dois modelos de mobilidade microscópicos originais: o *Intelligent Driver Model with Intersection Management* (IDM-IM) que trata do comportamento dos veículos em situações de cruzamentos e o *Intelligent Driver Model with Lane Changes* (IDM-LC) que regula o funcionamento de mudanças de faixas de modo a suportar ultrapassagens.

SUMO

Como já foi referido anteriormente, na realização do projecto desta dissertação, para gerar os cenários de mobilidade, foi utilizada a ferramenta *Simulation of Urban Mobility* (SUMO) [KHWR02]. Trata-se de uma aplicação *open source* que oferece bastantes funcionalidades, entre elas:

- A possibilidade de importar mapas topológicos da base de dados TIGER ou de ferramentas como o Vissim ou o Visum. Se preferível também é possível criar os mapas, de forma manual, por parte do utilizador. Na criação manual dos mapas topológicos, primeiramente o utilizador define um conjunto de nós (*nodes*) num ficheiro do tipo XML. De seguida, também num ficheiro XML, são definidos os caminhos (*edges*), que ligarão os nós e que poderão ser caracterizados pelo sentido em que os veículos circularão (p.e. do nó A para o nó B, ou vice-versa), número de faixas existentes, velocidade máxima permitida, entre outros. Podem também ser criados cruzamentos, entre caminhos, regulados pela regra da direita ou então pela existência de sinais luminosos. Finalmente, a ferramenta *sumo-netconvert*, recorrendo aos ficheiros criados anteriormente, cria a via de circulação dos veículos.
- Criar várias classes de veículos, que podem ser caracterizadas pelo seu comprimento, valores de aceleração e desaceleração, velocidade máxima e imperfeição do condutor.

A cada veículo, ou conjunto de veículos, é atribuída uma rota pré-definida, que é estabelecida informando quais os caminhos por onde o veículo terá de passar. Por fim, é ainda possível criar o veículo em si, associando-o a uma classe de veículo, a uma rota e a um instante de partida no tempo de simulação. Estes dados são também guardados num ficheiro do tipo XML. Conjugando o ficheiro que contém a via de circulação obtido anteriormente, e o ficheiro onde constam as classes de veículos, os veículos e as rotas, obtém-se o cenário de mobilidade.

O SUMO é uma aplicação que suporta micro-mobilidade, baseada no modelo de Krauß, apresentado na secção anterior, e como tal oferece um sistema livre de colisões, em que a velocidade de um veículo é determinada pela velocidade do veículo que o sucede. Assim, e em cenários de vias que apresentam mais do que uma faixa, esta ferramenta permite a realização de ultrapassagens: se um veículo, que por omissão circula na faixa mais à direita, estiver a deslocar-se com uma velocidade superior ao veículo que o sucede, e quase a alcançá-lo, caso a faixa da esquerda esteja livre, ele ultrapassa-o, regressando no final para a faixa onde circulava anteriormente.

O acesso à micro-mobilidade foi o principal factor para que esta ferramenta fosse a utilizada na geração dos cenários de mobilidade, descritos na secção seguinte. Outros factores como o fácil manuseamento da aplicação, a liberdade de parametrização na criação dos cenários, o facto dos mesmos serem compatíveis com o simulador de protocolos de rede ns-2, e até o grande suporte existente entre a comunidade, influenciaram positivamente a escolha desta aplicação.

3.3 Cenários de Mobilidade Gerados

A implementação dos cenários de mobilidade utilizados na realização do projecto desta dissertação pode ser dividida em três fases: inicialmente procedeu-se à criação da via de circulação dos veículos, recorrendo à ferramenta SUMO; de seguida foram criadas e caracterizadas as diferentes classes de veículos utilizadas na simulação; finalmente na terceira e última fase, e utilizando novamente a ferramenta SUMO, foram gerados os movimentos dos veículos, obtendo-se assim o cenário de mobilidade necessário para se realizarem as

simulações.

Relativamente à via de circulação dos veículos, e recorrendo ao comando *sumo-netconvert* da ferramenta SUMO, foi projectado um troço de auto-estrada, com 10 *km*, disposto em linha recta, composto por dois sentidos, de três faixas por cada sentido, com uma velocidade máxima estipulada em 133 *km/h*.

No que diz respeito aos veículos que compõem o cenário, e de modo a tentar recriar o conjunto de veículos que circulam numa auto-estrada, foram definidas três classes, sendo distinguíveis no seu comprimento, velocidade máxima e taxa de aceleração e desaceleração (Tabela 3.2):

- **Classe 1:** os veículos desta classe são caracterizados por uma velocidade máxima de 27.8 *m/s* (100 *km/h*), valores de aceleração e desaceleração de 3.6 *m/s²* e um comprimento de 4 *m*. Esta classe representa aproximadamente 60 % dos veículos que compõem o cenário.
- **Classe 2:** os veículos que pertencem a esta classe caracterizam-se por serem mais lentos que os da classe anterior, atingido apenas os 26.0 *m/s* (93.6 *km/h*), sendo parametrizados com valores de aceleração e desaceleração de 2.5 *m/s²* e 3.0 *m/s²* respectivamente, e um comprimento de 5 *m*. Esta classe corresponde aproximadamente a 25 % dos veículos que circulam na rede.
- **Classe 3:** por fim, os 15 % de veículos que restam no cenário de mobilidade fazem parte da classe 3, que caracteriza os veículos mais lentos. Nesta classes, os veículos atingem uma velocidade máxima de 20.0 *m/s* (72 *km/h*), parametrizados com aceleração de 1.5 *m/s²* e uma taxa de desaceleração de 2.0 *m/s²*, apresentando 8 *m* de comprimento (representando veículos pesados).

Após realizar a via de circulação e caracterizar as 3 classes de veículos, definiu-se a quantidade e a distribuição dos veículos que fazem parte da simulação. Foi definido um alcance rádio para os veículos de 1000 *m*, ou seja, segundo a Definição 3.1 apresentada no parágrafo seguinte, sempre que um veículo estiver a uma distância igual ou inferior a 1000 *m* de outro,

Tabela 3.2: Caracterização das classes de veículos implementadas.

	Comprimento (m)	Velocidade Máxima (m/s)	Aceleração (m/s^2)	Desaceleração (m/s^2)	% de Participação (ρ)
Classe 1	4	27.8	3.6	3.6	60
Classe 2	5	26.0	2.5	3.0	25
Classe 3	8	20	1.5	2.0	15

considera-se que existe uma ligação física entre ambos. Foram implementados quatro cenários de simulação de modo a representar quatro valores de densidade diferentes para a vizinhança de veículos. Em todos os cenários o comprimento da auto-estrada é mantido nos 10 km e o alcance rádio de cada nó a 1000 m . Consequentemente, são necessários, no mínimo, 10 saltos (*hops*) para que um veículo no início da auto-estrada consiga comunicar com outro que se encontre no final da mesma, valor este que se pretendeu manter em todos os cenários.

Definição 3.1. Noção de Vizinhança Física:

Seja N_x o conjunto de nós que estão dentro do alcance rádio do nó n_x . N_x representa o conjunto de vizinhos físicos de n_x , e os nós que constituem N_x estabelecem uma ligação física com n_x .

Cenário 4 vizinhos

No cenário de 4 vizinhos, de ora em diante denominado de Cen_{4v} , foram utilizados 80 veículos, divididos equitativamente pelos dois sentidos. Dos 40 veículos atribuídos para cada sentido, inicialmente são dispostos 20 veículos ao longo da auto-estrada, encontrando-se distribuídos de forma exponencial, com uma distância média de 500 m , de forma a se verificar uma densidade aproximada de 4 vizinhos (no mesmo sentido). Posteriormente, recorrendo novamente a uma distribuição exponencial, a cada 20 segundos (em média) entra um novo veículo no início da auto-estrada, em cada sentido.

Cenário 6 vizinhos

Por forma a aumentar o número de vizinhos mantendo a distância do troço de auto-estrada, houve a necessidade de se adicionar mais veículos à simulação. Assim, para se obter

um cenário com uma densidade média de 6 vizinhos, no mesmo sentido (Cen_{6v}), foram utilizados 120 veículos, também divididos de igual forma pelos dois sentidos. Neste cenário, são inicialmente dispostos 30 veículos em cada sentido, encontrando-se distribuídos com uma distância média de 333.3 m . No decorrer da simulação, e a cada 12 segundos (em média), é adicionado um novo veículo em cada sentido, no início da auto-estrada.

Cenário 8 vizinhos

O cenário de 8 vizinhos, (Cen_{8v}), é constituído por 160 veículos, divididos pelos dois sentidos. Por forma a se obter uma vizinhança média, em cada sentido, de 8 vizinhos, são distribuídos 40 veículos pela auto-estrada espaçados em média de 250 m , enquanto que os restantes são adicionados à simulação a cada 10 segundos.

Cenário 10 vizinhos

Por fim, o cenário que contempla mais veículos, e consequentemente um maior número de vizinhança, 10 vizinhos em média (Cen_{10v}), é formado por 200 veículos nos dois sentidos. Inicialmente, 50 veículos em cada sentido são uniformemente distribuídos pela auto-estrada, espaçados de aproximadamente 200 m , enquanto que os restantes entram na simulação em intervalos de, aproximadamente, 8 segundos.

Na Secção 1.2 observou-se a influência no desempenho do protocolo de encaminhamento OLSR num cenário com dois sentidos. Esta degradação é justificada pela utilização dos veículos existentes no sentido contrário para a difusão da topologia. Como a solução proposta nesta dissertação, para a difusão da topologia da rede, passa pela utilização dos veículos que apenas circulem no mesmo sentido, é importante verificar a duração das ligações físicas existentes entre todos os veículos que constituem o cenário.

A Figura 3.1 representa a duração das ligações físicas existentes entre os veículos nos 4 cenários de simulação utilizados. Em todos os cenários existem três picos que se destacam e que são justificados pelo cruzamento, durante a simulação, de veículos que circulem em sentidos opostos. Por exemplo, o terceiro pico, situado nos 50 segundos deve-se ao

cruzamento de veículos de classe 3, ou seja os veículos de classe mais lenta, e pode ser justificado por

$$T_{lig-pico3} = \frac{Raio \times 2}{v_{classe3} \times 2} \quad (3.4)$$

em que $v_{classe3} \times 2$ representa a velocidade relativa dos veículos dessa classe. Assim, pode-se afirmar que as durações de ligações com maior expressividade, situadas entre os 36 e os 50 segundos, são ligações físicas realizadas entre os veículos que circulam em sentidos opostos, e que se cruzam com velocidades relativas que vão desde os $72 \text{ km/h} \times 2 = 40 \text{ m/s}$, para o cruzamento de veículos de classe 3, e os $100 \text{ km/h} \times 2 = 55.6 \text{ m/s}$ para o cruzamento de veículos de classe 1.

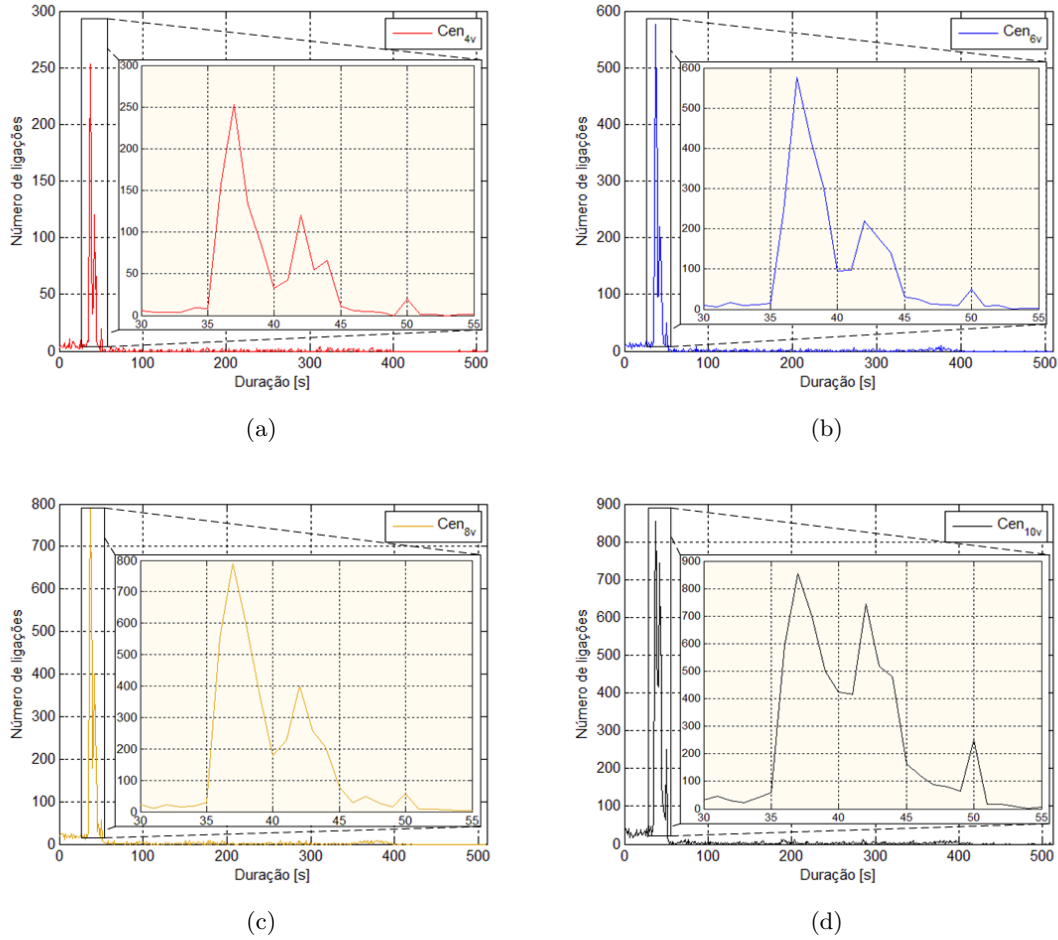


Figura 3.1: Duração das ligações físicas dos cenários Cen_{4v} (a), Cen_{6v} (b), Cen_{8v} (c) e Cen_{10v} (d).

A Figura 3.2 apresenta a função de distribuição acumulada (fda) da duração das ligações

físicas, dos 4 cenários utilizados. Aproximadamente 12% do total das ligações físicas têm duração inferior a 35 segundos, justificadas por ligações entre veículos que já se encontram distribuídos pela auto-estrada no início da simulação. Como referido anteriormente, grande parte das ligações, aproximadamente 73%, resultam do cruzamento de veículos de sentidos opostos, restando apenas aproximadamente 15% de ligações, que identificam ligações físicas entre veículos da mesma faixa. Estas ligações, que têm uma duração que vão desde os 50 segundos até os 512 segundos, **são as únicas ligações que serão utilizadas para difundir a topologia da rede do protocolo OLSR**, definidas por ligações âncora no capítulo seguinte, para assim aumentar o desempenho do mesmo.

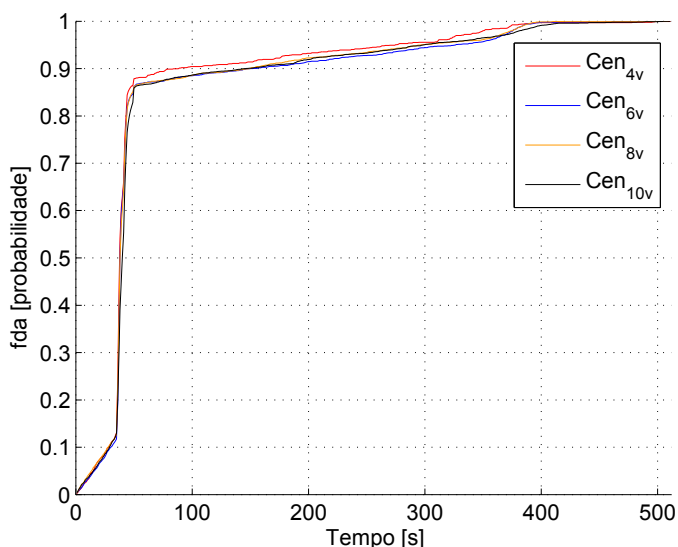


Figura 3.2: Função de distribuição acumulada da duração das ligações físicas dos 4 cenários de simulação.

Durante este capítulo foi apresentado o conceito de modelo mobilidade, juntamente com algumas características que fazem de um modelo de mobilidade o mais realista possível. Foram ainda conhecidos os diferentes modelos de mobilidade existentes, assim como as suas vantagens e fraquezas, e algumas aplicações geradoras de cenários de mobilidade, entre elas o SUMO, a ferramenta utilizada na realização do projecto desta dissertação. Por fim, foram ainda dados a conhecer os quatro cenários de simulação utilizados, sendo ainda iniciada a proposta de resolução do problema apresentado nesta dissertação, que será desenvolvida nos capítulos seguintes.

Capítulo 4

Identificação de Nós Âncora

As redes móveis não infra-estruturadas são caracterizadas pela frequente alteração de topologia. Este facto motiva a necessidade de cada nó manter o mais actualizado possível o seu conhecimento acerca da topologia da rede, pois o modo de funcionamento de alguns protocolos de encaminhamento com optimização de topologia, como o OLSR, baseiam o encaminhamento de dados nesta informação. A informação topológica é normalmente difundida em modo *broadcast* por nós seleccionados para o efeito (nós âncora) segundo um determinado critério. Normalmente utiliza-se o critério de eficiência de inundação, que consiste em seleccionar o menor número de nós âncora que permitem difundir a topologia a toda a rede.

Os métodos clássicos de selecção de nós âncora não contemplam como critério de selecção a duração da relação existente entre dois nós. No final do capítulo anterior, foi possível observar que os veículos que viajam no mesmo sentido tendem a manter as suas ligações durante mais tempo. Como tal, e com o intuito de optimizar o controlo topológico característico deste tipo de protocolos, é utilizado um método de eleição de nós âncora, que se baseia na duração das ligações existentes entre cada nó.

Este capítulo encontra-se dividido em quatro secções. Na primeira secção é introduzido o conceito de identificação de ligações estáveis, e a sua importância quando utilizado em algoritmos de encaminhamento baseados na optimização de topologia. Na segunda secção,

é apresentado o método utilizado para a identificação de ligações estáveis, enquanto que na terceira secção é expandido o conceito de estabilidade para um nó, apresentando-se então um algoritmo de agrupamento de nós baseado na estabilidade das ligações. Por fim, na última secção, é avaliado o desempenho do algoritmo nos cenários de simulação descritos no capítulo anterior.

4.1 Introdução

No final do Capítulo 3 observou-se que apenas 15% do número de ligações físicas representam a maioria das ligações existentes entre nós que viajam no mesmo sentido, caracterizando consequentemente as ligações mais duradouras. Estas ligações podem vir a ter um papel bastante importante nos protocolos de encaminhamento baseados em optimização de topologia para redes *ad-hoc*.

Este tipo de protocolos de encaminhamento, durante o seu funcionamento, definem um conjunto de ligações entre nós que, para além de realizarem controlo de topologia, são também utilizadas para encaminhamento. Em cenários de elevada mobilidade, como em cenários de auto-estrada, é bastante importante que o conjunto de ligações seleccionado tenha em conta a sua duração, podendo assim oferecer as seguintes garantias:

- Aumento da estabilidade do protocolo face à alta mobilidade dos nós. As ligações com duração inferior a um certo limiar, nunca deverão ser seleccionadas como rotas de encaminhamento, pois têm maior probabilidade de se quebrarem.
- Diminuição de situações de incoerência no processo de encaminhamento, que é baseado na informação topológica da rede. Para tal, as ligações mais duradouras são as únicas que deverão ser escolhidas para realizar o controlo de topologia.

Por forma a excluir as 85% de ligações com duração inferior a 50 segundos é utilizado um algoritmo desenvolvido em 2005 por Oliveira et. al [OBP05], tendo sido melhorado em 2009 [Oli09]. O modo de funcionamento deste algoritmo pode ser dividido em duas fases. Numa primeira fase são identificadas as ligações que apresentem uma duração acima do limiar pré-definido, definindo-as como ligações estáveis. Com base na identificação destas

ligações, é depois possível fazer uma selecção de nós, por forma a criar um grupo de difusão de controlo de topologia que, apesar de não garantir a cobertura total da rede, oferece mais estabilidade na disseminação da informação.

4.2 Identificação de Ligações Consoante a Duração

Os protocolos de encaminhamento utilizados em redes *ad-hoc* móveis não têm noção de ligação física. Assim, para tomar conhecimento da sua vizinhança, e por forma a poder estabelecer uma ligação lógica, um nó recorre ao envio/recepção periódico de mensagens em modo *broadcast*, também conhecidas por *beacons*¹. A duração de uma ligação estabelecida entre dois nós é caracterizada pelo número de *beacons* recebidos ininterruptamente. Por exemplo, se um nó n_a receber de uma forma sucessiva 15 *beacons* do nó n_b , então n_a pode afirmar que a sua ligação lógica com n_b tem a duração de 15 períodos de *beacons*.

Antes de se iniciar a descrição do algoritmo é necessário fazer a distinção entre vizinhança física apresentada no Capítulo 3 e o conceito de vizinhança lógica, tendo em conta que agora existe um meio lógico de identificação de vizinhos físicos. São ainda definidos alguns conceitos e assumidas ligações bidireccionais entre dois nós.

Definição 4.1. Noção de Vizinhança Lógica:

Seja N_x o conjunto de nós de quem o nó n_x recebe *beacons* durante um determinado intervalo de tempo. N_x representa o conjunto de nós vizinhos lógicos de n_x .

Note-se que a Definição 3.1 se refere às condições necessárias para o estabelecimento de comunicação física entre dois nós. A noção de vizinhança lógica é mais restrita, pois dois nós podem ser vizinhos (segundo a Definição 3.1), mas, devido à não recepção de *beacons* num dado intervalo de tempo, a relação de vizinhança lógica pode não se verificar. Daqui em diante, a menos que seja dito algo em contrário, a utilização do termo vizinho refere-se à Definição 4.1.

¹Tramas de envio periódico normalmente utilizadas pelos protocolos de encaminhamento para descoberta de vizinhança.

O envio periódico de *beacons* é realizado com uma frequência de $1/T_B$. Considera-se o instante $t_i(n_y)$ em que um nó(n_a) recebe o primeiro *beacon* transmitido pelo seu vizinho n_y , estabelecendo-se uma ligação lógica entre os nós (n_a e n_y). Define-se, de seguida, o conceito de estabilidade da ligação lógica.

Definição 4.2. Estabilidade da Ligação Lógica:

Um nó n_a que receba *beacons* de um nó vizinho n_y possui um determinado valor de estabilidade η com esse vizinho. A estabilidade $\eta(n_y)$ afere a duração da relação de vizinhança entre os nós. O valor de $\eta(n_y)$, determinado pelo nó n_a no instante temporal t , é dado pela expressão $\eta(n_y) = 1 + (t - t_i(n_y)) \text{ div } T_B$, onde $a \text{ div } b$ representa a operação de divisão inteira entre a e b .

Os nós mantêm uma tabela de *beacons* (também denominada tabela de vizinhos lógicos) que descreve as ligações lógicas desse nó com os seus nós vizinhos. Os nós vizinhos são representados por cada um dos registos da tabela de *beacons*, que nesta fase do algoritmo é constituída por:

- O endereço do nó vizinho que envia o *beacon* ($n_y \in N_x$).
- Um campo temporário utilizado pelo algoritmo contendo o valor da estabilidade da ligação com esse vizinho ($\eta(n_y)$), sendo actualizado no instante em que é executado o algoritmo.
- O instante $t_i(n_y)$ em que foi recebido o primeiro *beacon*.
- O intervalo de tempo $T_O(n_y)$ em que o registo ainda é válido

O número de registos contidos na tabela é finito, e, após atingir o limite máximo admissível, não é possível adicionar novos registos à tabela.

Tendo em conta a Definição 4.2, é possível então determinar, quando é que uma ligação que apresente uma certa duração pode ser considerada como uma ligação estável.

Definição 4.3. Estabilidade de ligação:

Um nó n_x possui uma ligação estável com um dos seus nós vizinhos $n_y \in N_x$ no instante

t se:

$$\eta(n_y) \geq k_{est},$$

onde k_{est} é um limiar de estabilidade previamente definido, e $\eta(n_y)$ representa o valor da estabilidade da ligação.

Adaptando a Definição 4.3 aos cenários de auto-estrada estudados no final do Capítulo 3, ao ser definido $k_{est} = 50$, apenas 15% das ligações serão consideradas estáveis sendo as restantes consideradas ligações instáveis. Tomemos por exemplo um cenário mais simples. Na Figura 4.1 é apresentada uma rede composta por 6 nós onde as linhas a tracejado representam as relações de vizinhança existentes.

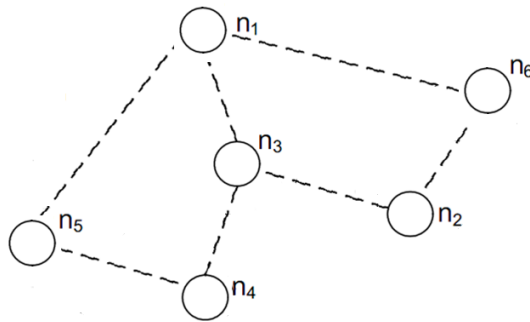


Figura 4.1: Rede *ad hoc* constituída por 6 nós ($\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$).

A Tabela 4.1 apresenta valores hipotéticos para a tabela de *beacons* do nó n_3 representado na Figura 4.1. A tabela apresenta três registros (linhas) relativos aos nós vizinhos n_1, n_2 e n_4 . O nó n_3 possui uma ligação lógica com o vizinho n_1 há 65 múltiplos do período de *beacon* (T_B) sem que o temporizador associado ao registo da tabela desse vizinho tenha expirado. Desta forma, diz-se que a ligação mais estável do nó n_3 é a que mantém com o nó n_1 . Se, por exemplo, considerarmos esta tabela de *beacons* retirada de algum nó que pertença aos cenários de auto-estrada, então a única ligação realmente estável que o nó n_3 tem com algum dos seus vizinhos é com o nó n_1 , pois as ligações com o nó n_2 e o nó n_4 ainda não atingiram $k_{est} = 50$.

Definido o método de identificação das ligações mais duradouras, que agora podem ser de-

Tabela 4.1: Exemplo do conteúdo da tabela de *beacons* do nó n_3 representado na Figura 4.1 no instante $t = 102.5s$, sendo $T_B = 1s$.

$N_3 = \{n_1, n_2, n_4\}$	$\eta(n_y), \forall n_y \in N_3$	$t_i(n_y)$	$T_O(n_y)$
n_1	65	37.2	k_1
n_2	30	72.3	k_2
n_4	2	100.1	k_4

signadas de ligações estáveis, é possível então excluir as ligações existentes entre veículos que circulam em sentidos opostos. Para tal, basta parametrizar o limiar de estabilidade, de forma a que os valores de estabilidade estejam directamente relacionados com a velocidade relativa dos veículos que circulam em sentidos opostos. Para $k_{est} = 50$ todas as ligações que apresentem uma duração inferior a 50 segundos serão consideradas ligações instáveis.

É importante ainda referir que, sendo um algoritmo que se baseia no número de *beacons* recebidos por forma a caracterizar uma ligação, apresenta uma grande vantagem em cenários de elevada mobilidade. Nos cenários gerados, a parametrização do limiar de estabilidade com o valor de 50 segundos é suficiente para distinguir os veículos que circulam em sentidos opostos. No entanto, se o algoritmo for aplicado, por exemplo num cenário citadino, o limiar de estabilidade vai necessariamente ser mais elevado, fazendo com que a caracterização das ligações relativamente à sua estabilidade seja mais demorado.

4.3 Selecção de Nós Âncora

Caracterizadas as ligações em estáveis ou instáveis, é possível estudar a segunda etapa deste algoritmo, que tem como objectivo agrupar os nós existentes na rede (em grupos denominados de Grupos de *Broadcast* (GB)), baseado na estabilidade da ligações de um nó para com os seus nós vizinhos. Dentro de cada GB existe um nó eleito como nó âncora, também denominado de Líder de Grupo de *Broadcast* (LGB), que será responsável pela difusão das mensagens topológicas, dentro de cada GB, e pelo encaminhamento. Este algoritmo de agrupamento de nós, baseado na estabilidade das ligações, objectiva a selecção do número de nós âncora, de forma a definir um conjunto de nós. Apesar de não garantirem cobertura total da rede, o conjunto de nós deve ser utilizados como um grupo

de nós de difusão de mensagens topológicas, devido às suas características particulares de mobilidade. Desta forma, não só se obtém uma redução do tráfego *broadcast* gerado na rede, como se pretende aumentar a eficiência da difusão face à mobilidade dos nós.

Definição 4.4. Líder do Grupo de *Broadcast* (LGB):

Todo o nó n_a que pertence ao conjunto \mathcal{N} dos nós da rede pode eleger um nó vizinho denominado líder do grupo de *broadcast*. O nó LGB eleito pelo nó n_a é representado por $\xi(n_a)$.

Definição 4.5. Grupo de *Broadcast* (GB):

Dado o conjunto de nós $N = \{n_1, n_2, \dots, n_{g-1}, n_g\}$, a condição

$$\forall n_i \in N : \exists_1 n_{lgb} = \xi(n_i)$$

impõe que todos os nós pertencentes a N escolham o mesmo nó LGB (nó n_{lgb}). Nestas condições um grupo de *broadcast* é definido através do conjunto $GB = N \cup \{n_{lgb}\}$.

Transpondo o conceito de estabilidade de ligação apreendido na secção anterior, define-se o conceito de estabilidade associado a um nó.

Definição 4.6. Nó estável:

Um nó n_x é estável se possui pelo menos um vizinho $n_y \in N_x$ com o qual possui uma ligação estável.

Um nó é denominado instável sempre que a condição expressa na Definição 4.6 não se verifica. Por outras palavras, um nó é instável quando não possuir nenhuma ligação estável com qualquer dos seus nós vizinhos (a duração da ligação lógica expressa em múltiplos do período T_B é sempre inferior a k_{est}). O significado de estabilidade do nó permite caracterizar parcialmente a mobilidade do nó, pois permite identificar as relações de mobilidade relativa com os seus nós vizinhos. Note-se que, caso dois nós móveis possuam uma ligação estável, esta pode manter-se, mesmo quando os nós têm um valor de mobilidade elevada, desde que os dois nós estejam em posições espaciais que permitam a troca de *beacons* entre si.

Como pretendido, a selecção do conjunto de nós LGB tem em conta a duração/estabilidade das ligações de cada nó para com os outros. Antes de se proceder à explicação mais detalhada do algoritmo, tomemos como exemplo a rede estudada na secção anterior. Ao ser executado o algoritmo de selecção de nós âncora no nó n_3 , no instante $t = 102.5$ segundos (Tabela 4.1), e com um limiar de estabilidade $k_{est} = 50$, o nó n_3 elege como seu nó LGB o único nó que pode ser considerado estável, ou seja, o nó n_1 . Nesta etapa do algoritmo, cada entrada na tabela de *beacons* fica com mais um campo que indica qual o nó LGB de cada nó vizinho de n_3 .

A rede apresentada na Figura 4.1 é redesenhada na Figura 4.2, por forma a identificar os GB existentes e os nós que foram eleitos como nós LGB. As linhas a tracejado continuam a representar as relações de vizinhança existentes, enquanto que as linhas a cheio definem os GB. Os nós LGB são representados por círculos pretos.

A Tabela 4.2 apresenta o conteúdo da tabela de *beacons* do nó n_3 , desta vez com a indicação dos nós LGB eleitos pelos vizinhos de n_3 . De notar que o nó n_1 se auto-elege como LGB, enquanto que os nós n_2 e n_4 elegem os LGB n_6 e n_5 , respectivamente, de acordo com um conjunto de regras que serão apresentadas no parágrafo seguinte. Como já foi referido anteriormente, o nó n_3 possui uma ligação lógica com o vizinho n_1 há 65 períodos de *beacon*, o que torna o nó n_1 o nó vizinho mais estável de n_3 . Na verdade, e tendo em conta a Definição 4.6, o nó n_1 é o único nó vizinho estável de n_3 .

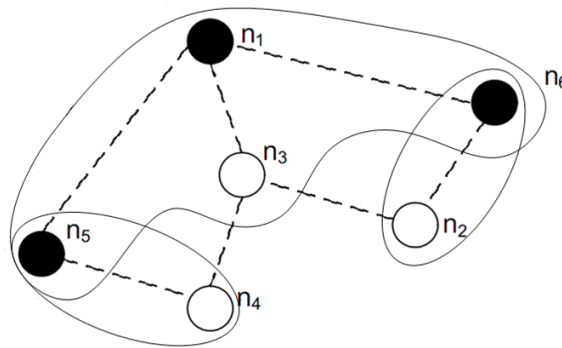


Figura 4.2: Rede *ad hoc* constituída por 6 nós ($\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$), onde existem 3 GB, e os nós n_1, n_5 e n_6 são LGB.

Tabela 4.2: Exemplo do conteúdo da tabela de *beacons* do nó n_3 representado na Figura 4.2 no instante $t = 102.5s$, sendo $T_B = 1s$, após eleição de nós LGB.

$N_3 = \{n_1, n_2, n_4\}$	$\eta(n_y), \forall n_y \in N_3$	$\xi(n_y), \forall n_y \in N_3$	$t_i(n_y)$	$T_O(n_y)$
n_1	65	n_1	37.2	k_1
n_2	30	n_6	72.3	k_2
n_4	2	n_5	100.1	k_4

Para a criação dos diferentes GB é necessário que todos os nós executem um algoritmo de eleição do seu próprio LGB. O algoritmo de eleição é distribuído e não necessita de qualquer outro tipo de troca de informação além da contida nos diferentes *beacons* enviados pelos nós vizinhos. Cada nó executa o algoritmo de eleição antes de enviar o *beacon*, a fim de enviar informação actualizada acerca do seu estado. No algoritmo, são utilizados os endereços dos nós, que se assume serem inteiros atribuídos univocamente a cada um dos nós. Antes de enviar um novo *beacon*, um nó n_a elege o seu LGB, aplicando as seguintes regras:

- **R1:** quando n_a é instável, não elege nenhum LGB. Caso contrário, aplica as regras descritas a seguir;
- **R2:** quando nenhum dos nós n_y vizinhos de n_a se encontra eleito como LGB de um GB, o nó n_a elege como seu LGB um vizinho n_y com o menor endereço de entre aqueles com o qual possui as ligações com maior valor de estabilidade de ligação;
- **R3:** quando n_a já se encontra eleito como LGB por um dos seus nós vizinhos, e todos os nós LGB vizinhos possuem um endereço superior ao seu, então o nó n_a auto-elege-se como LGB;
- **R4:** quando n_a não é eleito por nenhum dos seus vizinhos, e existe pelo menos um nó vizinho n_y que já se encontra eleito LGB, então o nó n_a elege o nó n_y como seu LGB. Quando existe mais do que um nó vizinho n_y eleito LGB, n_a elege o vizinho com o menor endereço.

Foi utilizado o Algoritmo 4.1 para eleição do LGB. O algoritmo toma como parâmetros de entrada a informação contida na tabela de *beacons*. Na linha 1, a função "encontra_maximo_eta_na_tabela_de_beacons()" devolve o maior valor de estabilidade de ligação

parâmetros de entrada	$: N_a, \eta(n_y) (\forall n_y \in N_a), \xi(n_y) (\forall n_y \in N_a), t_i(n_y) (\forall n_y \in N_a)$
parâmetros de saída	$: \xi(n_a)$

```

1   $\eta_{max} \leftarrow \text{encontra\_maximo\_}\eta\_\text{na\_tabela\_de\_beacons}()$ 
2  endereço  $\leftarrow \text{MAX\_INT}$ 
3   $\xi_{aux} \leftarrow -1$ 
4  limiar_transiente  $\leftarrow 1$ 
5  if nó_estável( $n_a$ ) then                                     /* R1 - se este nó for estável */
6      for cada nó vizinho ( $n_y \in N_a$ ) do
7          | insere_em_lista_ordenada( $\xi(n_y)$ , lista_LGB) /* menor endereço na cabeça da lista */
8          |
9          if ( $n_a$  é LGB) then
10             | insere_em_lista_ordenada( $n_a$ , lista_LGB)
11         for cada elemento  $\xi_y \in \text{lista\_LGB}$  do /* retira o elemento na cabeça da lista */
12             | for cada nó vizinho ( $n_y \in N_a$ ) do
13                 | if ( $n_y = \xi_y$ ) e nó_estável( $n_y$ ) then /* R4 - elege um nó vizinho que já é LGB */
14                     |  $\xi_{aux} \leftarrow n_y$ 
15             |
16             if ( $\xi_{aux} \neq -1$ ) then /* acabou de eleger um LGB */
17                 | break
18             if ( $n_a = \xi_y$ ) then /* R3 - auto-eleição */
19                 |  $\xi_{aux} \leftarrow n_a$ 
20                 | break
21         if ( $\xi_{aux} = -1$ ) then /* R2 - elege um nó vizinho sem ser LGB */
22             | for cada nó vizinho ( $n_y \in N_a$ ) do
23                 | if ( $(\eta_{max} - \eta(n_y) - \text{limiar\_transiente} \leq 0)$  e ( $n_y < \text{endereço}$ )) then
24                     | endereço  $\leftarrow n_y$ 
25                     |  $\xi_{aux} \leftarrow n_y$ 
26             |
27         if ( $\xi_{aux} = -1$ ) then
28             |
29      $\xi(n_a) = \xi_{aux}$ 

```

Algoritmo 4.1: Algoritmo de eleição do Líder de Grupo de Broadcast (LGB) do nó genérico n_a .

$\eta(x_y)$ atualizado a partir da tabela de *beacons* para ser aplicado na regra R2. Na linha 2, a variável "endereço" é iniciada com o maior inteiro possível. Caso n_a seja estável com algum dos seus vizinhos, o algoritmo começa por criar uma lista ordenada de forma crescente com o endereço dos LGB eleitos pelos seus nós vizinhos (linhas 6 e 7), a qual pode também conter o nó n_a , caso tenha sido eleito LGB por algum dos seus nós vizinhos (linhas 8 e 9). Se existirem nós vizinhos já eleitos LGB, n_a elege como LGB o nó vizinho LGB que possui o menor endereço, o que é realizável porque os elementos retirados na lista na linha 10 encontram-se ordenados de forma crescente de endereços. Note-se, no entanto, que, caso algum vizinho tenha já eleito n_a como LGB, o nó n_a auto-elege-se LGB (linhas 17 a 18). Caso não existam nós vizinhos eleitos LGB, n_a elege o seu nó vizinho com maior valor de estabilidade de ligação (linhas 21 a 25). Caso haja mais do que uma

ligação com o maior valor de estabilidade, aplica-se um critério de desempate, elegendo como LGB o vizinho que possui o menor endereço. O algoritmo utiliza ainda a constante "limiar_transiente=1" para considerar (no critério da linha 23) valores de estabilidade de ligações lógicas que ainda poderão valer η_{max} durante o período de tempo T_B .

A Figura 4.3(a) ilustra uma rede *ad hoc* móvel onde o Algoritmo 4.1 é executado. No instante inicial o nó n_1 é LGB, tendo sido eleito pelos nós n_3 , n_5 , n_6 e por ele próprio (auto-eleição). Esta eleição forma um GB composto pelos nós $\{n_1, n_3, n_5, n_6\}$. Suponha-se que os nós n_2 e n_4 se movem para a vizinhança dos nós (n_3, n_5) e (n_3, n_6) , criando noutro instante as ligações lógicas representadas na Figura 4.3(b). Quando as ligações entre os nós n_2 e n_6 e entre n_4 e n_5 se tornam estáveis, os nós n_5 e n_6 são eleitos LGB pelos nós n_4 e n_2 , respectivamente, criando dois GB. Tal como se verifica no exemplo, o algoritmo origina uma árvore de grupos de *broadcast*, a qual é centrada no nó LGB possuindo o menor endereço. A Figura 4.4 apresenta a árvore de grupos de *broadcast* vista pelo nó n_1 , a qual é composta pelo nó raiz (n_1) e três ramos representados pelos nós n_5 , n_6 e n_3 . Cada ramo da árvore é constituído por um nó n_i , cujo LGB é o nó antecessor ($\xi(n_5) = \xi(n_6) = \xi(n_3) = n_1$). As folhas da árvore (nós n_2, n_4 e n_3) representam os nós que não são eleitos LGB.

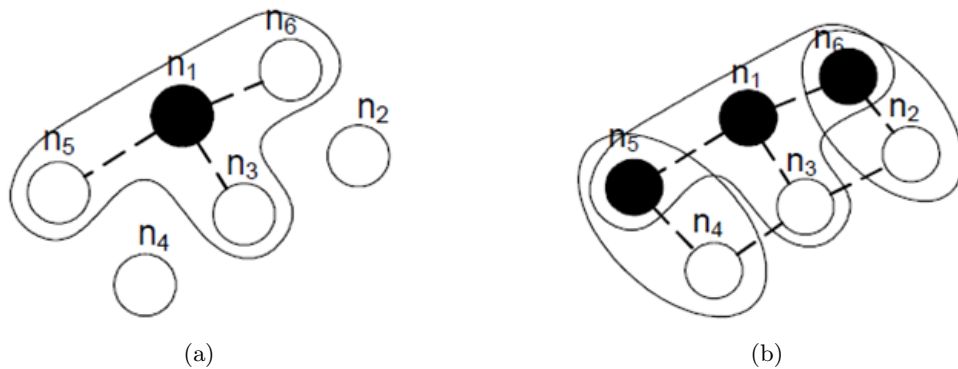


Figura 4.3: Rede móvel *ad hoc* constituída por 6 nós ($\mathcal{N} = \{n_1, n_2, n_3, n_4, n_5, n_6\}$) em diferentes instantes temporais: (a) os nós n_2 e n_4 não possuem ligações lógicas; (b) os nós n_2 e n_4 possuem ligações lógicas com os nós n_3 e n_6 , e n_3 e n_5 respectivamente.

O algoritmo origina um núcleo de nós (*backbone*) composto por nós LGB. Os nós LGB podem estar ligados num núcleo que assegura a cobertura total da rede, tal como acon-

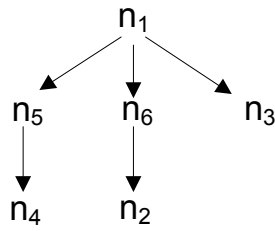


Figura 4.4: Árvore de grupos de *broadcast* do nó n_1 representado no exemplo da Figura 4.3(b).

tece entre os LGB n_1, n_5, n_6 da Figura 4.2. Ou seja, basta que os nós LGB transmitam uma mensagem topológica para que todos os nós da rede o recebam. No entanto, existem situações em que o grupo de nós LGB eleitos não cobrem todos os nós existentes na rede. Como tal, é importante relembrar que o principal objectivo da utilização deste algoritmo não é a criação de um conjunto de nós capazes de cobrir todos os nós da rede, mas sim distinguir as ligações estáveis das ligações instáveis, adaptando a ideia de estabilidade aos algoritmos de encaminhamento baseados na optimização de topologia.

Exemplifica-se uma situação em que o conjunto de nós LGB não cobre todos os nós da rede. Na Figura 4.5, um GB composto pelos nós n_7 e n_8 move-se de forma a que o nó n_7 se torne vizinho do nó n_3 . O nó n_7 continua com ligação ao nó n_8 , e como tal continua a elegê-lo como seu LGB aplicando a regra R4. A Figura 4.6 representa a árvore de grupos de *broadcast* neste caso (vista pelo nó n_1). Neste caso, o conjunto de nós LGB existentes na rede não possui nós LGB a interligar dois GB, pois o nó n_8 não se encontra ligado a mais nenhum nó LGB.

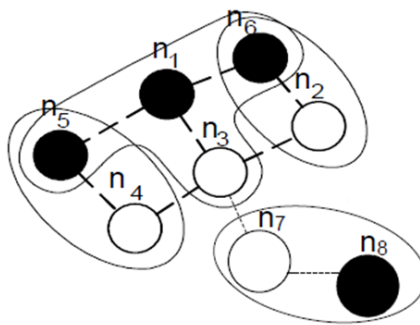


Figura 4.5: Exemplo de fusão de um GB.

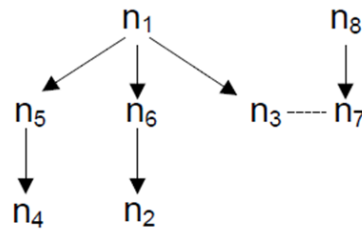


Figura 4.6: Árvore de grupos de *broadcast* no caso pior em que a conectividade entre os GB $\{n_1, n_3, n_5, n_6\}$ e $\{n_7, n_8\}$ são separados por dois nós não LGB (nós n_3 e n_7).

O desempenho do algoritmo de agrupamento depende da estabilidade da rede. O período de transmissão do *beacon* deve ser seleccionado de acordo com os valores de mobilidade dos nós. Caso uma grande percentagem dos nós seja estável, o algoritmo detecta-os, e os GB criados pelo algoritmo podem ser utilizados por outros algoritmos para diminuir a carga total da rede. Para manter a constituição dos grupos actualizada numa rede que apresenta mobilidade muito elevada, o algoritmo poderá exibir custos demasiado elevados: as rápidas alterações observadas na rede implicam um aumento da frequência de transmissão dos *beacons* para que o estado das ligações lógicas com os nós vizinhos possa ser identificado com o menor grau de incoerência. O aumento da frequência de transmissão dos *beacons* origina um aumento na carga da rede, a qual causa outros problemas, tais como o aumento das colisões entre tramas e a degradação de *throughput*. Consequentemente, é importante caracterizar a rede em termos da sua mobilidade máxima, pois a frequência de transmissão dos *beacons* deve ser adequada à velocidade máxima dos nós, por forma a que as ligações com os nós vizinhos sejam correctamente detectadas com valores de carga longe da situação de tempestade de *broadcast*.

4.4 Avaliação de Desempenho

Esta sub-secção apresenta resultados da avaliação de desempenho do algoritmo de agrupamento dos nós anteriormente apresentado (Algoritmo 4.1). São também apresentadas algumas estatísticas referentes à quantidade de GB criados durante a simulação e o número médio de nós que compõe um GB, assim como o número médio de vizinhos de um nó.

Como objecto de estudo foram utilizados os quatro cenários de simulação anteriormente

apresentados, tendo sido o algoritmo parametrizado com uma frequência média de transmissão de *beacons* de 1 Hz ($T_B = 1\text{ s}$), com um atraso/avanço no envio do *beacon* que pode atingir no máximo 2.5% do período T_B , o qual é amostrado uniformemente no intervalo $[0.975, 1.025]\text{ s}$. O parâmetro k_{est} utilizado na função "nó_estável" foi inicializado a 50, que como se observou no final da Secção 3.3, se refere ao período mais longo em que duas viaturas que circulam em sentidos opostos mantêm uma ligação física. O parâmetro T_O foi inicialmente parametrizado a 2.5 segundos, tendo sido depois modificado (até ser feita nova referência, os resultados apresentados foram obtidos com $T_O = 2.5$ segundos). A Tabela 4.3 apresenta a duração média de actividade de um nó na rede, ou seja, o tempo que o nó demora a percorrer a auto-estrada.

Tabela 4.3: Duração média de actividade de um nó, nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

	Cen_{4v}	Cen_{6v}	Cen_{8v}	Cen_{10v}
Tempo Médio de Participação (s)	361.775	358.7167	359.1875	369.315

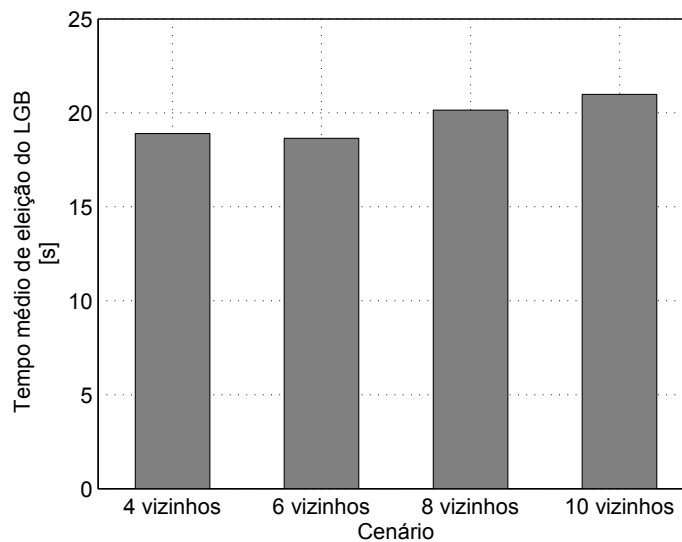


Figura 4.7: Tempo médio de eleição do LGB para os cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

A Figura 4.7 apresenta o tempo médio que cada um dos nós mantém eleito o mesmo LGB. Este tempo é a média dos tempos que os nós mantêm um determinado LGB eleito. Tendo

em conta o tempo de vida útil de um nó, apresentado na Tabela 4.3, cada nó mantém o mesmo LGB eleito aproximadamente durante 6% do tempo total que o nó se encontra activo. À medida que a densidade de nós aumenta, é de notar também um ligeiro aumento no tempo médio de eleição do LGB.

A Figura 4.8 apresenta os resultados do número médio teórico e real de vizinhos por nó, bem como a dimensão média de cada GB agrupado pelo Algoritmo 4.1 para os 4 cenários de simulação. O valor teórico do número de nós apresentado na figura representa o valor médio de vizinhos (4, 6, 8 e 10 no mesmo sentido) utilizado como referência na Secção 3.3, para determinar a parametrização do modelo de mobilidade. Como se pode observar, nos 4 cenários de simulação o número médio real de vizinhos é bastante próximo do valores teoricamente pretendidos. É também possível verificar que o número médio de nós que compõem os GB nunca é inferior a 75% do número médio real de nós vizinhos, o que indica que os GB apenas são constituídos por nós que circulam no mesmo sentido, pois os que viajam em sentido contrário foram considerados instáveis.

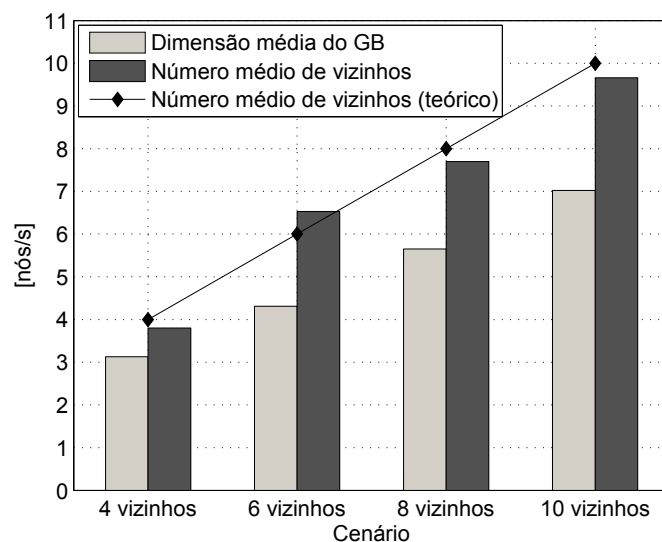


Figura 4.8: Número médio de vizinhos por nó (teórico e real), e dimensão média do GB obtida com o Algoritmo 4.1 aplicado ao cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

A Figura 4.8 evidencia ainda que o número de nós vizinhos aumenta à medida que a densidade de nós também aumenta, característica que é detectada pelo algoritmo de agru-

pamento de nós, pois também se verifica um aumento na dimensão média dos GB com o aumento da densidade de nós. Por outro lado, dado que valores mais elevados de densidade significam um maior número de vizinhos, a probabilidade de um nó ser estável durante mais tempo aumenta à medida que a densidade de nós aumenta, pois a probabilidade de perder todas as ligações estáveis de um nó é inferior. Este comportamento pode ser observado na Figura 4.9, onde a percentagem de tempo em que um nó é considerado instável vai diminuindo ligeiramente à medida que a densidade de nós aumenta.

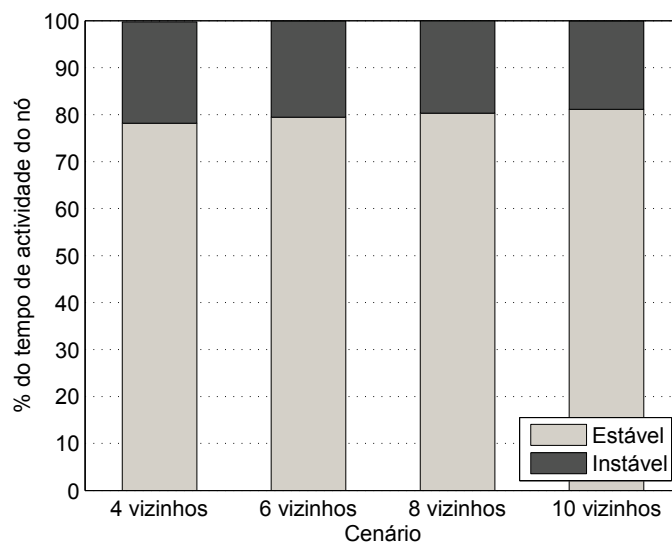


Figura 4.9: Percentagem das durações da estabilidade e da instabilidade em tempo por nó nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

O aumento da densidade de nós não se reflecte apenas na dimensão média de cada GB. Como se pode observar pela Figura 4.10 o número médio de GB existentes, formados pelo Algoritmo 4.1, também é influenciado pela variação do número de vizinhos de cada nó. À medida que a densidade de nós vai aumentando, o número médio de GB também aumenta, permitindo manter um equilíbrio entre a dimensão dos GB (Figura 4.8) e a densidade de nós existentes no cenário.

Por fim, a Figura 4.11 compara o histograma da duração das ligações lógicas criadas através do algoritmo de agrupamento de nós, com o histograma das ligações físicas efectivamente existentes. É importante que a duração das ligações lógicas seja idêntica à duração

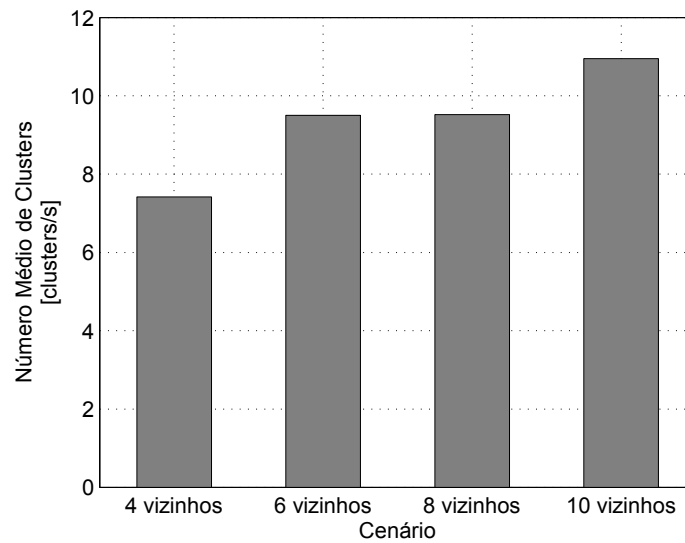


Figura 4.10: Número médio de GB nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

das ligações físicas, pois o conceito de estabilidade utilizado pelo algoritmo, baseia-se nas durações das ligações lógicas, por forma a realizar a correcta identificação de nós âncora. Relativamente ao cenário Cen_{4v} (Figura 4.11(a)) a duração das ligações físicas e das ligações lógicas são praticamente idênticas. Este resultado deriva do facto de se ter parametrizado $k_{est} = 50$, o que fez com que o algoritmo de agrupamento não considerasse como nós estáveis os nós que circulam no sentido contrário.

No que diz respeito aos restantes cenários de simulação, houve a necessidade de se parametrizar T_O com outros valores, para além dos 2.5 segundos previamente estabelecidos. Em relação ao cenário de 6 vizinhos, Cen_{6v} (Figura 4.11(b)), se for utilizado $T_O = 2.5$ segundos nota-se o aparecimento de uma grande quantidade de novas ligações lógicas com uma duração entre os 3 e os 15 segundos, que não existem como ligações físicas. A razão do aparecimento das novas ligações lógicas advém do facto de, com o aumento da densidade de nós a probabilidade de colisões aumenta, aumentando também a probabilidade de se perderem *beacons*, permitindo que ligações lógicas já estabelecidas se quebrem inadvertidamente. Como tal, para corrigir esta situação, foram testados novos valores de T_O . Para o cenário de 6 vizinhos, foi suficiente adiar T_O mais um segundo, para 3.5 segundos, o que faz com que as durações das ligações lógicas sejam suficientemente prolongadas para que

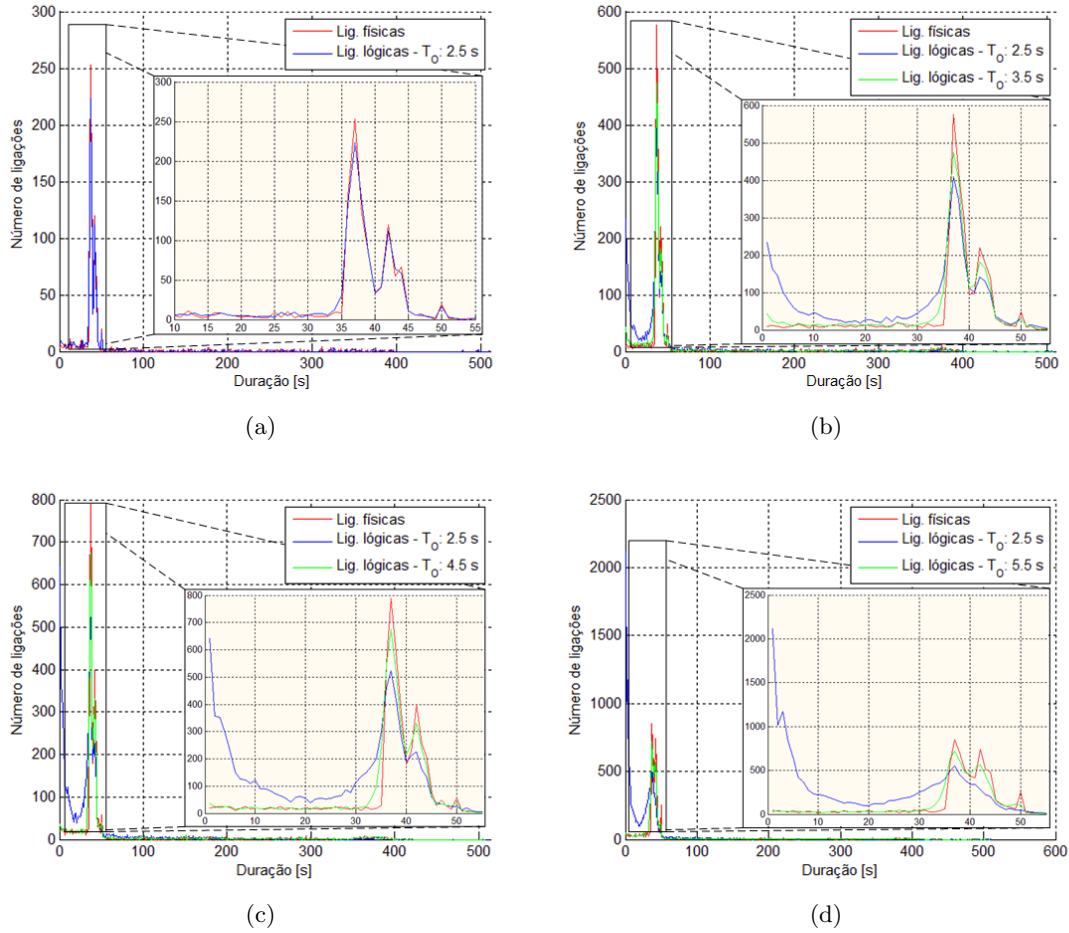


Figura 4.11: Comparação entre as durações das ligações físicas e ligações lógicas dos cenários Cen_{4v} (a), Cen_{6v} (b), Cen_{8v} (c) e Cen_{10v} (d).

não sejam declaradas quebradas, devido à perda de beacons por colisão. No cenário de 8 vizinhos (Figura 4.11(c)), é mais acentuado o aparecimento de novas ligações lógicas com duração inferior a 20 segundos, obtidas com a utilização de $T_O = 2.5$ segundos. Como a parametrização $T_O = 3.5$ segundos não apresentou resultados satisfatórios, houve a necessidade de se aumentar T_O para 4.5 segundos, por forma a se obter uma precisão elevada da visão real da rede. De igual modo, no cenário de 10 vizinhos (4.11(d)) houve a necessidade de se parametrizar T_O com 5.5 segundos, e assim fazer coincidir o histograma da durações das ligações físicas e das ligações lógicas.

A necessidade de se adaptar o parâmetro T_O deriva da elevada quantidade de tráfego a 1-salto, no qual origina mais colisões e situações de congestão ao nível MAC. Como tal, é

possível um ajuste automático deste parâmetro, a partir da densidade de vizinhos aferida pelo comprimento da tabela de vizinhos. Adaptando este parâmetro de forma correcta é possível então de verificar que as ligações lógicas são mapeadas com sucesso nas ligações físicas.

Durante este capítulo foi demonstrada a importância de caracterizar as ligações entre cada nó, baseada na sua duração. A utilização de ligações mais estáveis, por parte de algoritmos de encaminhamento baseados na optimização de topologia, pode garantir um aumento de estabilidade do protocolo, face à elevada mobilidade dos nós. Foi apresentado um algoritmo de agrupamento de nós baseado na estabilidade das ligações, que permite não só a identificação das ligações estáveis, como também a criação de GB entre nós que viajam no mesmo sentido. Desta forma, o encaminhamento torna-se mais estável e, o controlo de topologia característico de protocolos como o OLSR mais leve, em termos de tráfego *broadcast*. Assim, é possível agora adaptar o protocolo de encaminhamento OLSR a este algoritmo de agrupamento de nós, tarefa que é apresentada no capítulo seguinte.

Capítulo 5

Encaminhamento com Otimização de Topologia

Os protocolos de encaminhamento com otimização de topologia, baseiam em parte o seu modo de funcionamento nos protocolos de encaminhamento *link-state* pro-activo. A grande diferença está no modo como é realizada a difusão da topologia da rede. Nos protocolos *link-state* pro-activo a difusão é realizada por todos os nós da rede, o que provoca uma elevada carga de tráfego *broadcast*. Nos protocolos de encaminhamento com otimização de topologia, como o OLSR, esta difusão é realizada por um conjunto de nós (denominados no Capítulo 4 por nós âncora), seleccionados com o intuito de reduzir o número de retransmissões de mensagens topológicas, e consequentemente uma diminuição do tráfego *broadcast*.

A grande debilidade deste protocolo aplicado a redes veiculares, conforme observado no Capítulo 1 com a existência de tráfego nos dois sentidos, está no conjunto de heurísticas que regem a eleição dos nós âncora (denominados no protocolo OLSR por nós *Multipoint Relay* (MPR)): não contemplam a duração das ligações entre cada nó, cingindo-se apenas à disponibilidade em termos de energia do aparelho e à garantia de eleição de nós suficientes para cobrir toda a rede. Com a adaptação do algoritmo apresentado no capítulo anterior, pretende-se melhorar o desempenho do protocolo OLSR, modificando o conjunto de regras de eleição dos nós âncora, considerando a estabilidade das ligações existentes

entre cada nó e, se for preciso, sacrificando a cobertura total da rede.

Este capítulo encontra-se dividido em duas secções. Na primeira secção é apresentado detalhadamente o protocolo de encaminhamento adoptado nesta dissertação, o OLSR, com especial atenção para o método de eleição de nós âncora, característica fundamental deste protocolo. Na segunda e última secção são apresentadas as alterações propostas ao método de eleição de nós âncora do OLSR, adaptando o algoritmo de agrupamento de nós estudado no capítulo anterior.

5.1 *Optimized Link State Routing (OLSR)*

5.1.1 Visão Global

O *Optimized Link State Routing* (OLSR) [JMC⁺01] é um protocolo de encaminhamento distribuído, com optimização de topologia, que apresenta algumas características de encaminhamento pro-activo, desenvolvido para redes sem fios que apresentem uma arquitectura *ad-hoc*. A principal característica deste protocolo está na selecção de um conjunto de nós denominados de *Multipoint Relays* (MPR) que são responsáveis pela difusão das mensagens topológicas na rede, ao contrário do originalmente definido pelo conceito de encaminhamento *link-state*, em que todos os nós difundiam esta informação. Como tal, o mecanismo de difusão de mensagens topológicas é optimizado (razão para a existência do termo *Optimized* no nome do protocolo), na medida em que este tipo de nós fornecem um mecanismo de difusão eficiente, já que o número de retransmissões é reduzido, diminuindo assim a carga de tráfego *broadcast* existente na rede [QLV01].

O protocolo OLSR, para além das mensagens de controlo de topologia características do encaminhamento *link-state*, não gera qualquer tipo de tráfego extra sempre que exista a quebra ou a formação de uma nova ligação. A periódica disseminação da topologia da rede mantém os nós informados das alterações verificadas na rede, podendo também ser considerado um protocolo tolerante à perda de pacotes, facto que se verifica com regularidade em redes sem fios. Trata-se de um protocolo que suporta redes de bastante densidade, devido sobretudo à utilização dos nós MPR, verificando-se uma crescente optimização do

funcionamento do protocolo à medida que o número de nós da rede vai aumentando, em comparação com o encaminhamento *link-state* padrão.

O tipo de encaminhamento realizado pelo OLSR pode ser considerado um encaminhamento salto-a-salto. Cada nó utiliza a informação que contém acerca da vizinhança a 1-salto para realizar o envio de pacotes, deixando para o próximo nó que recebe o pacote a escolha de qual o próximo destinatário do mesmo. Esta característica é bastante importante, já que a mobilidade é um factor a ter em conta em cenários de redes sem fios: a rota que previamente estaria definida para um pacote pode ter sido alterada devido a uma modificação da disposição dos nós.

Multipoint Relays

A ideia da utilização de nós MPR, como já foi referido, é a de minimizar a quantidade de tráfego *broadcast* existente na rede, reduzindo o número de retransmissões de mensagens duplicadas. Cada nó n da rede selecciona um conjunto de nós entre os seus vizinhos a 1-salto, responsáveis pela retransmissão das mensagens de topologia. Todos os outros vizinhos de n que não façam parte deste conjunto apenas recebem a mensagem, processam-na, mas não a retransmitem. Para tal, cada nó MPR mantém um conjunto de vizinhos denominados de MPR *Selectors*, e sempre que uma mensagem de topologia vinda de um nó pertencente a este conjunto é recebida, sabe-se que deve ser retransmitida, pois o nó receptor foi escolhido como MPR pelo nó emissor.

Cada nó escolhe o seu conjunto de nós MPR de entre todos os seus vizinhos a 1-salto que possam garantir, em termos de alcance rádio, a total cobertura dos nós a 2-saltos. O conjunto de nós MPR do nó n , denominado de $MPR(n)$, é considerado um sub-conjunto da vizinhança de n que perfaz a seguinte condição: qualquer nó na vizinhança a 2-saltos de n , N_2 , tem de ter uma ligação bidireccional com o conjunto $MPR(n)$. Quanto menor for $MPR(n)$ maior é a optimização do protocolo, pois exigirá um menor número de retransmissões de mensagens. Como exemplo, representa-se na Figura 5.1 uma rede sem fios em forma de estrela, em que o nó n escolhe 4 dos seus 8 vizinhos a 1-salto como nós MPR,

pois é o mínimo necessário para cobrir toda a vizinhança a 2-saltos. Consequentemente, o número de retransmissões necessárias para informar todos os nós vizinhos a 2-saltos de n é reduzido.

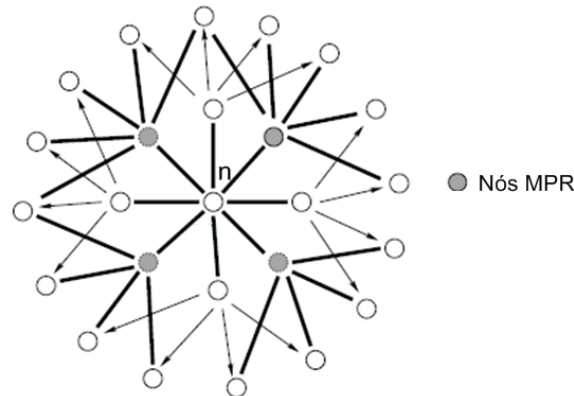


Figura 5.1: Rede sem fios com a utilização de nós MPR.

Os nós MPR não servem apenas para disseminar informação topológica da rede. O encaminhamento deve também ser feito pelos nós MPR. Para tal, cada nó transmite periodicamente o conjunto de nós que o seleccionaram como nó MPR. Com base nesta informação, cada nó calcula e actualiza a sua rota para cada nó de destino conhecido, através do seu nó MPR, construindo assim sempre que possível, um encaminhamento através de nós MPR. O facto de existir uma ligação bidireccional entre os nós MPR e os seus MPR *Selectors*, garante que a transferência de dados se possa realizar em ligações bidireccionais.

5.1.2 Funcionamento do Protocolo

De uma forma geral, o modo de funcionamento do protocolo OLSR pode ser definido em 4 etapas. Primeiro, é necessário que um nó descubra quem são os seus nós vizinhos. De seguida, e tendo conhecimento de quem são os seus vizinhos a 1-salto, procede-se à eleição dos nós MPR. Assim que forem seleccionados nós MPR suficientes há que informar os restantes nós, podendo, por fim, ser calculada a tabela de encaminhamento de modo a se proceder ao envio de tráfego de dados.

Conhecimento da Vizinhança

A primeira etapa do funcionamento do protocolo OLSR passa pelo conhecimento da topologia local, ou seja, da vizinhança a 1-salto. Cada nó precisa de detectar os seus vizinhos a 1-salto com os quais tem uma ligação directa e bidireccional. Sendo as redes sem fios um tipo de redes bastante característico em termos de qualidade de ligação, é necessário que todas as ligações sejam verificadas em ambos os sentidos, para que possam ser consideradas ligações bidireccionais.

Seguindo o modo de funcionamento do encaminhamento *link-state* padrão, cada nó toma conhecimento da vizinhança através do envio periódico (T_B) de mensagens HELLO, difundidas em modo *broadcast*, contendo a informação sobre os seus actuais vizinhos (que podem ser classificados como *Symmetric*, MPR ou *Not Neighbor*) e os estados das respectivas ligações (*Unspecified*, *Asymmetric* (Unidireccional), *Symetric* (Bidireccional) e *Lost*). Estas mensagens são recebidas por todos os nós a 1-salto, não sendo reencaminhadas. Uma mensagem HELLO é constituída, entre outros campos, por:

- Uma lista dos endereços dos vizinhos com a qual exista uma ligação bidireccional válida;
- Uma lista dos endereços dos vizinhos com a qual exista uma ligação, mas que ainda não tenha sido validada como bidireccional: se o nó receptor de uma mensagem HELLO encontrar nesta lista o seu próprio endereço, então pode considerar que tem uma ligação bidireccional válida com o nó emissor.

Outra informação bastante importante e necessária para a eleição dos nós MPR, é o campo *willingness* (campo da mensagem HELLO), cuja função será abordada na secção seguinte.

Esta troca de mensagens HELLO permite não só o conhecimento da vizinhança a 1-salto, mas também a 2-saltos. É com base na informação trocada por estas mensagens, e depois guardada numa tabela de vizinhança durante um certo período de tempo, que cada nó selecciona os seus nós MPR, sendo depois indicados na mensagem HELLO com o seguinte par [vizinho MPR, ligação bidireccional]. A troca de mensagens HELLO, permite que um

nó construa a sua tabela de MPR *Selectors*, que contém todos os seus nós vizinhos que o seleccionaram como nó MPR.

Eleição de Nós MPR

Cada nó elege o seu conjunto de nós MPR, independentemente da escolha feita pelos outros nós da rede. Como já foi referido anteriormente, este conjunto de nós é calculado por forma a que esteja garantida a cobertura de todos os nós vizinhos a 2-saltos. Antes de ser apresentado o algoritmo que representa o método de eleição de nós MPR, existem algumas definições que devem ser consideradas.

Definição 5.1. Nó n_a :

O nó que está a proceder à eleição do seu conjunto de nós MPR.

Definição 5.2. *Willingness*:

O parâmetro *willingness*(n) (interpretado por voluntariedade) está associada a cada nó, e representa o seu livre interesse em retransmitir tráfego. Um nó que parametrize a sua *willingness* como WILL_NEVER nunca deverá ser eleito como um nó MPR, nem pertencer a uma rota. Caso um nó se apresente com *willingness* WILL_ALWAYS deverá sempre ser seleccionado como nó MPR. Por defeito, um nó deverá apresentar a sua *willingness* como WILL_DEFAULT.

Definição 5.3. Subconjunto N_a :

N_a representa o subconjunto de vizinhos a 1-salto do nó n_a , que apresentem uma ligação bidireccional com este. Qualquer nó que pertença a este subconjunto é representado por n_y .

Definição 5.4. Subconjunto N_{2a} :

N_{2a} representa o subconjunto de vizinhos a 2-saltos do nó n_a , excluindo:

- os vizinhos a 2-saltos que apenas são alcançáveis por membros de N que apresentem *willingness* = WILL_NEVER;
- o próprio nó n_a ;
- todos os vizinhos simétricos, ou seja, os vizinhos no qual existe uma ligação bidireccional para o nó n_a .

Qualquer nó que pertença a este subconjunto é representado por n_{2y} .

Definição 5.5. *Alcançabilidade*(n_y, n_a):

A quantidade de nós pertencentes a N_{2a} que ainda não estão cobertos por nenhum nó MPR de n_a , e que são alcançáveis através do nó n_y , é denominada de alcançabilidade do nó n_y .

Definição 5.6. *Degree*(n_y):

Degree(n_y) (também designado por associatividade) é definido como sendo o número de vizinhos simétricos de n_y , excluindo todos os membros de N , e ainda o nó que realiza a eleição do conjunto de MPR, n_a .

O Algoritmo 5.1 apresenta o método de eleição de nós MPR. São tomados como parâmetros de entrada as vizinhanças N_a e N_{2a} . A eleição do conjunto de nós MPR do nó n_a é feita tendo em conta as seguintes regras:

- **R1:** Todos os nós vizinhos a 1-salto de n_a que apresentem *willingness* = WILL_ALWAYS são automaticamente eleitos para o conjunto de nós MPR (linhas 1 a 4).
- **R2:** Calcula-se *degree*(n_y) para todos os nós vizinhos a 1-salto de n_a (linha 21). Apesar de ser a regra R2, só será aplicada durante a execução da regra R4.2.
- **R3:** São eleitos como nós MPR, todos os nós vizinhos a 1-salto que sejam os **únicos** capazes de ligar n_a a qualquer vizinho a 2-saltos de n_a (linhas 6 a 9).
- **R4:** Enquanto existirem n_{2y} que ainda não estão cobertos por qualquer nó MPR:
 - **R4.1:** Para cada nó vizinho n_y calcular a sua alcançabilidade (linha 12).
 - **R4.2:** Elege-se como nó MPR o nó n_y que apresentar maior valor de *willingness* de entre todos os nós N_a que não apresentarem uma alcançabilidade nula. Em caso de empate deve-se eleger o nó que fornecer maior alcançabilidade. Novamente em caso de empate, é eleito o nó que apresentar um maior valor de *degree* (linhas 13 a 22).

À medida que as regras vão sendo aplicadas, são retirados do subconjunto N_{2a} , os nós vizinhos a 2-saltos que vão estando cobertos pelos nós vizinhos a 1-salto, eleitos com nós

MPR.

```

    parâmetros de entrada      :  $N_a, N_{2a}$ 
    parâmetros de saída       :  $MPR(n_a)$ 

1  for cada nó vizinho a 1-salto ( $n_y \in N_a$ ) do
2      if  $willingness(n_y) = WILL\_ALWAYS$  then                /* Regra R1 */
3          insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
4          remove_nós( $N_{2a}, n_y$ )
5
6  for cada nó vizinho a 2-saltos ( $n_{2y} \in N_{2a}$ ) do
7      if  $n_a$  só alcançar  $n_{2y}$  por  $n_y$  then                /* Regra R3 */
8          insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
9          remove_nós( $N_{2a}, n_y$ )
10
11 while  $\exists n_{2y} \in N_{2a}$  que ainda não estão cobertos por pelo menos 1 nó MPR do /* Regra R4 */
12     calcular_alcancabilidade( $n_a, n_y$ )                    /* Regra R4.1 */
13     if  $willingness(n_y) > willingness(n), n \in N_a$  then /* Regra R4.2 */
14         insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
15     else
16         if  $willingness(n_y) = willingness(n), n \in N_a$  then
17             if  $alcancabilidade(n_y, n_a) > alcancabilidade(n, n_a), n \in N_a$  then
18                 insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
19             else
20                 if  $alcancabilidade(n_y, n_a) = alcancabilidade(n, n_a), n \in N_a$  then
21                     if  $degree(n_y) > degree(n), n \in N_a$  then /* Regra R2 */
22                         insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
23
24
25
26
27     remove_nós( $N_{2a}, n_y$ )
28

```

Algoritmo 5.1: Algoritmo de eleição do conjunto de nós MPR do nó genérico n_a .

É necessário proceder-se à reeleição de nós MPR sempre que se verificarem os seguintes acontecimentos:

- Uma mudança na topologia da rede a 1-salto, com o aparecimento ou a quebra de uma ligação bidireccional.
- Uma mudança no subconjunto de vizinhos a 2-saltos, com o conhecimento de uma nova ligação bidireccional.

Depois de n_a elaborar o seu conjunto de nós MPR, é necessário informar os seus vizinhos a 1-salto, no próximo envio periódico da mensagem HELLO. Assim, cada nó vizinho que receber a mensagem pode construir a sua tabela de MPR *Selectors*. Por exemplo, se o

nó n_y tiver sido escolhido por n_a para ser seu nó MPR, então n_y vai adicionar na sua tabela de MPR *Selectors* o endereço do nó n_a , juntamente com um número de sequência associado ao nó n_a .

Disseminação de Informação de Controlo de Topologia

Por forma a ser possível a construção das tabelas de encaminhamento, cada nó difunde em modo de *broadcast* mensagens de controlo da topologia específicas, denominadas de *Topology Control (TC) messages*. As mensagens TC podem ser equiparadas às mensagens *link-state* características deste tipo de encaminhamento, no entanto são apenas difundidas e retransmitidas pelos nós previamente seleccionados como nós MPR, enquanto o seu tempo de vida associado não expirar.

Uma mensagem TC é enviada periodicamente, por cada nó da rede, por forma a declarar o seu conjunto de MPR *Selector*. Ou seja, a mensagem TC contém a lista de todos os vizinhos que seleccionaram o nó que emite a mensagem como nó MPR, juntamente com o número de sequência associado a este conjunto de MPR *Selector*. A lista de endereços do conjunto MPR *Selector* pode por vezes ser incompleta, desde que periodicamente seja emitida com todos os endereços do conjunto MPR *Selector*. Esta difusão frequente da mensagem TC permite aos nós que a receberem, a possibilidade de elaborarem a sua tabela de encaminhamento. Um nó que tenha um conjunto MPR *Selector* vazio, i.e., que não foi eleito por nenhum nó vizinho como nó MPR, pode não gerar nenhuma mensagem TC.

A periodicidade do envio de mensagens TC está dependente da modificação, ou não, do conjunto MPR *Selector* desde a última vez que foi recebido. Quando ocorre uma alteração neste conjunto, a próxima mensagem TC deverá ser enviada mais cedo do que o agendado, mas sempre após um tempo mínimo pré-determinado, desde que enviou a última mensagem TC. Caso este intervalo de tempo já tenha sido ultrapassado, a mensagem é enviada imediatamente, sendo as seguintes mensagens TC enviadas periodicamente até nova alteração na topologia da rede, nomeadamente, no seu conjunto MPR *Selector*.

Cada nó da rede mantém uma tabela, também denominada de tabela de topologia, na qual grava a informação acerca da topologia da rede obtida pelas mensagens TC. A tabela de topologia apresenta os seguintes campos:

- ***Destination Address:*** Endereço do nó de destino, potencialmente alcançável a 1-salto pelo emissor da mensagem TC.
- ***Last Address:*** Endereço do último nó por onde passa o tráfego antes de atingir o nó de destino. Este nó é o emissor da mensagem TC, ou seja, o nó MPR do nó de destino.
- ***Sequence Number:*** O número de sequência do conjunto MPR *Selector*.
- ***Time:*** Período de validação, no qual indica que a informação é válida durante este período de tempo, devendo depois ser removida da tabela de topologia.

Por forma a guardar a informação recebida na recepção de uma mensagem TC, o nó deve realizar o seguinte procedimento:

1. Se existir alguma entrada na tabela de topologia cujo campo *Last Address* corresponda ao endereço do emissor da mensagem TC, e o número de sequência do conjunto MPR *Selector* for maior do que o proveniente na mensagem, então não existem alterações a fazer, pois a informação já existente no nó está mais actualizada do que a recebida.
2. Se existir alguma entrada na tabela de topologia cujo campo *Last Address* corresponda ao endereço do emissor da mensagem TC, e o número de sequência do conjunto MPR *Selector* for menor do que o proveniente na mensagem, então a entrada existente na tabela é apagada.
3. Para cada endereço existente no conjunto MPR *Selector* da mensagem TC recebida aplicam-se as seguintes condições:
 - (a) Se existir alguma entrada na tabela de topologia cujo campo *Destination Address* corresponda ao endereço do MPR *Selector*, e o campo *Last Address* for

igual ao endereço do emissor da mensagem TC, então o período de validação é renovado.

- (b) Caso contrário, é adicionada uma nova entrada na tabela de topologia.

Elaboração da Tabela de Encaminhamento

A tabela de encaminhamento é uma peça fundamental nos protocolos de encaminhamento, pois permite o encaminhamento de tráfego para outros nós da rede e, como tal, deve ser calculada com a máxima precisão e estar sempre actualizada. Sempre que um nó recebe uma mensagem TC, processa-a, guardando os denominados "pares de ligação" na forma [último nó, nó de destino], onde os "nós de destino" são os endereços existentes na mensagem TC e o "último nó" o endereço do emissor da mensagem TC. A tabela de encaminhamento é construída através desta informação, consultando os pares de ligação em ordem descendente. Por exemplo, na Figura 5.2 é apresentada uma rede em que os nós MPR são representados por círculos pretos e as formas quadradas representam a origem e o destino no envio de tráfego. Os pares de ligação indicados dizem respeito à informação que cada nó MPR envia nas suas mensagens TC (neste caso apenas é considerada uma ordem de difusão do destino até à origem). Quando o nó de Origem calcular a entrada na tabela de encaminhamento respeitante ao nó de Destino, vai encontrar o par (E, Destino), indo de seguida encontrar o par (D, E), e assim sucessivamente, até por fim encontrar o par (A, B) em que A é seu vizinho a 1-salto, dando assim por concluído o cálculo da rota.

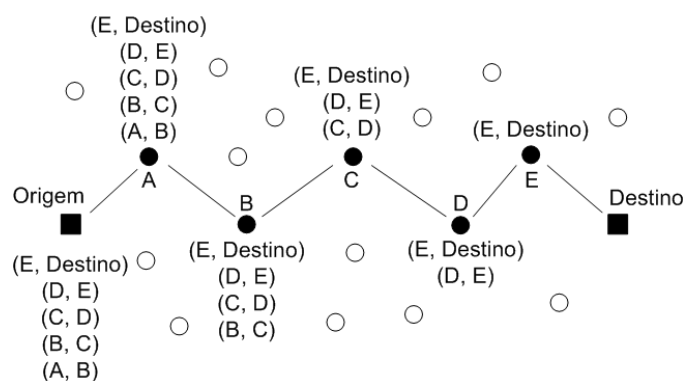


Figura 5.2: Elaboração de uma rota através de informação existente na tabela de topologia.

Por forma a calcular o caminho óptimo, os nós que participam no encaminhamento esco-

lhem apenas os pares de ligação que apresentarem um caminho mais curto até ao nó de destino. Esta selecção pode ser elaborada dinamicamente e utilizando uma baixa quantidade de recursos locais. Os números de sequência são utilizados para detectar pares de ligação que ao longo do tempo, e devido às alterações da topologia, se tornaram inválidos.

A tabela de encaminhamento apresenta a seguinte estrutura:

- ***Destination Address***: Entrada semelhante à existente na tabela de topologia, que contém o endereço de um nó de destino.
- ***Next Address***: O endereço do próximo nó para onde deverá ser enviado o tráfego com destino ao *Destination Address*.
- ***Interface Address***: Um nó pode apresentar várias interfaces, sendo cada uma identificada por um endereço de interface. Este campo não é relevante para o assunto em questão, e como tal não será feita mais nenhuma referência.
- ***Distance***: Distância, em número de saltos, do nó de origem até ao *Destination Address*.

Cada entrada é gravada na tabela de encaminhamento, indexada pelos endereços dos nós de destino em que as rotas sejam conhecidas. Qualquer nó de destino cuja rota esteja incompleta ou quebrada não pode ser adicionado na tabela.

A tabela de encaminhamento é baseada na informação existente na tabelas de vizinhança (a 1-salto e a 2-saltos) e na tabela de topologia. Assim, se alguma destas tabelas sofrer alterações, a tabela de encaminhamento é novamente elaborada de modo a actualizar a informação lá existente. Mais propriamente, a tabela de encaminhamento é recalculada sempre que se verificar uma alteração na vizinhança relacionada com ligações bidireccionais, ou se o período de validação de uma entrada na tabela de topologia expirar. O recalcular da tabela de encaminhamento não origina disseminação de informação extra.

O Algoritmo 5.2 apresenta o método de elaboração da tabela de encaminhamento. São tomados como parâmetros de entrada as tabelas de vizinhança, tanto de 1-salto como de

2-saltos, e a tabela de topologia. Por forma a (re)calcular a tabela de encaminhamento, o nó deverá realizar o seguinte procedimento:

- **Passo 1:** Todas as entradas da tabela de encaminhamento são apagadas (linha 3).
- **Passo 2:** As novas entradas são guardadas na tabela começando pelos nós de destino que sejam vizinhos a 1-salto ($h = 1$). Para cada entrada da tabela de vizinhança, cuja ligação não seja unidireccional e o tempo de validação ainda não tenha expirado, é gerada uma nova entrada na tabela de encaminhamento, em que o campo *Destination Address* e o campo *Next Address* têm obviamente o mesmo endereço. A distância é colocada a 1 (linhas 4 a 6). O método utilizado no algoritmo para adicionar novas entradas na tabela de encaminhamento (`adiciona_entrada_tabela_de_encaminhamento`) tem os seguintes parâmetros de entrada: o nó que faz o processamento, o endereço do nó de destino, o endereço do próximo nó e a distância.
- **Passo 3:** De seguida são inseridas na tabela as rotas para os nós que se encontrem a mais do que 1 salto de distância ($h + 1$). O procedimento apresentado de seguida é executado, ciclicamente, para consecutivos valores de h , iniciado com $h = 1$. O ciclo é interrompido quando não for registada uma nova entrada em qualquer iteração (linhas 8 a 24).
 - Para cada entrada na tabela de topologia, se o campo *Destination Address* não corresponder a nenhum campo *Destination Address* já existente na tabela de encaminhamento e, o campo *Last Address* da tabela de topologia corresponder a algum *Destination Address* existente na tabela de encaminhamento com uma distância igual a h , então é registada uma nova entrada na tabela de encaminhamento com os seguintes parâmetros:
 - * *Destination Address* é colocado com o valor do campo *Destination Address* da tabela de topologia;
 - * *Next Address* é colocado com o valor do campo *Next Address* da tabela de encaminhamento, cujo *Destination Address* seja igual ao *Last Address* mencionado anteriormente;
 - * *Distance* é colocada a $h + 1$.

- **Passo 4:** Depois de calculada a tabela de encaminhamento, as entradas na tabela de topologia que não foram usadas no cálculo de rotas devem ser removidas, por forma a ocupar menos espaço de memória, e também para evitar a existência de múltiplas rotas.

```

parâmetros de entrada      :  $N_a, N_{2a}, \text{Tabela\_de\_topologia}(n_a)$ 
parâmetros de saída       :  $\text{Tabela\_de\_encaminhamento}(n_a)$ 

1 tempo_de_validacao  $\leftarrow$  encontra_tempo_de_validacao( $n_a, n_y$ )
2 entrada_A, entrada_B  $\leftarrow$  Entrada na tabela de encaminhamento
3 apagar_tabela_de_encaminhamento( $n_a$ )                                /* Passo 1 */
4 for  $n_y \in N_a$  do                                                    /* Passo 2 */
5   if estado_da_ligação( $n_y, n_a$ ) = Bidireccional  $\&\&$  tempo_de_validacao  $\geq$  tempo_actual then
6   |   adiciona_entrada_tabela_de_encaminhamento( $n_a, n_y, n_y, 1$ )
7   |
8 for  $n_{2y} \in N_{2a}$  do                                                /* Passo 3 */
9   if  $n_y$  de  $n_a = n_y$  de  $n_{2y}$   $\&\&$  estado_da_ligação( $n_y, n_a$ ) = Bidireccional then
10  |   if  $!(n_y$  de  $n_a = n_y$  de  $n_{2y}$   $\&\&$  willingness( $n_y$ ) = WILL_NEVER) then
11  |   |   if  $!(n_y$  de  $n_a = n_{2y}$  de  $n_{2y}$   $\&\&$  estado_da_ligação( $n_y, n_a$ ) = Bidireccional) then
12  |   |   |   adiciona_entrada_tabela_de_encaminhamento( $n_a, n_{2y}, n_y, 2$ )
13  |   |
14  |   |
15  |   |
16 for  $h=2; h++$  do
17   for Entradas da Tabela de Topologia de  $n_a$  do
18   |   entrada_A =
19   |   procura_entrada_por_Dest_Adr(entrada_da_tabela_de_topologia::Destination_Address)
20   |   entrada_B =
21   |   procura_entrada_por_Dest_Adr(entrada_da_tabela_de_topologia::Last_Address)
22   |   if  $\nexists$  entrada_A  $\&\&$   $\exists$  entrada_B  $\&\&$  entrada_B::Distance =  $h$  then
23   |   |   adiciona_entrada_tabela_de_encaminhamento( $n_a,$ 
24   |   |   |   entrada_da_tabela_de_topologia::Destination_Address, entrada_B::Next_Address,
25   |   |   |    $h + 1$ )
26   |   if Não existiu nenhuma entrada na tabela de encaminhamento then
27   |   |   break
28   |
29   |
30   |
31   |
32   |
33   |
34   |
35   |
36   |
37   |
38   |
39   |
40   |
41   |
42   |
43   |
44   |
45   |
46   |
47   |
48   |
49   |
50   |
51   |
52   |
53   |
54   |
55   |
56   |
57   |
58   |
59   |
60   |
61   |
62   |
63   |
64   |
65   |
66   |
67   |
68   |
69   |
70   |
71   |
72   |
73   |
74   |
75   |
76   |
77   |
78   |
79   |
80   |
81   |
82   |
83   |
84   |
85   |
86   |
87   |
88   |
89   |
90   |
91   |
92   |
93   |
94   |
95   |
96   |
97   |
98   |
99   |
100  |
101  |
102  |
103  |
104  |
105  |
106  |
107  |
108  |
109  |
110  |
111  |
112  |
113  |
114  |
115  |
116  |
117  |
118  |
119  |
120  |
121  |
122  |
123  |
124  |
125  |
126  |
127  |
128  |
129  |
130  |
131  |
132  |
133  |
134  |
135  |
136  |
137  |
138  |
139  |
140  |
141  |
142  |
143  |
144  |
145  |
146  |
147  |
148  |
149  |
150  |
151  |
152  |
153  |
154  |
155  |
156  |
157  |
158  |
159  |
160  |
161  |
162  |
163  |
164  |
165  |
166  |
167  |
168  |
169  |
170  |
171  |
172  |
173  |
174  |
175  |
176  |
177  |
178  |
179  |
180  |
181  |
182  |
183  |
184  |
185  |
186  |
187  |
188  |
189  |
190  |
191  |
192  |
193  |
194  |
195  |
196  |
197  |
198  |
199  |
200  |
201  |
202  |
203  |
204  |
205  |
206  |
207  |
208  |
209  |
210  |
211  |
212  |
213  |
214  |
215  |
216  |
217  |
218  |
219  |
220  |
221  |
222  |
223  |
224  |
225  |
226  |
227  |
228  |
229  |
230  |
231  |
232  |
233  |
234  |
235  |
236  |
237  |
238  |
239  |
240  |
241  |
242  |
243  |
244  |
245  |
246  |
247  |
248  |
249  |
250  |
251  |
252  |
253  |
254  |
255  |
256  |
257  |
258  |
259  |
260  |
261  |
262  |
263  |
264  |
265  |
266  |
267  |
268  |
269  |
270  |
271  |
272  |
273  |
274  |
275  |
276  |
277  |
278  |
279  |
280  |
281  |
282  |
283  |
284  |
285  |
286  |
287  |
288  |
289  |
290  |
291  |
292  |
293  |
294  |
295  |
296  |
297  |
298  |
299  |
300  |
301  |
302  |
303  |
304  |
305  |
306  |
307  |
308  |
309  |
310  |
311  |
312  |
313  |
314  |
315  |
316  |
317  |
318  |
319  |
320  |
321  |
322  |
323  |
324  |
325  |
326  |
327  |
328  |
329  |
330  |
331  |
332  |
333  |
334  |
335  |
336  |
337  |
338  |
339  |
340  |
341  |
342  |
343  |
344  |
345  |
346  |
347  |
348  |
349  |
350  |
351  |
352  |
353  |
354  |
355  |
356  |
357  |
358  |
359  |
360  |
361  |
362  |
363  |
364  |
365  |
366  |
367  |
368  |
369  |
370  |
371  |
372  |
373  |
374  |
375  |
376  |
377  |
378  |
379  |
380  |
381  |
382  |
383  |
384  |
385  |
386  |
387  |
388  |
389  |
390  |
391  |
392  |
393  |
394  |
395  |
396  |
397  |
398  |
399  |
400  |
401  |
402  |
403  |
404  |
405  |
406  |
407  |
408  |
409  |
410  |
411  |
412  |
413  |
414  |
415  |
416  |
417  |
418  |
419  |
420  |
421  |
422  |
423  |
424  |
425  |
426  |
427  |
428  |
429  |
430  |
431  |
432  |
433  |
434  |
435  |
436  |
437  |
438  |
439  |
440  |
441  |
442  |
443  |
444  |
445  |
446  |
447  |
448  |
449  |
450  |
451  |
452  |
453  |
454  |
455  |
456  |
457  |
458  |
459  |
460  |
461  |
462  |
463  |
464  |
465  |
466  |
467  |
468  |
469  |
470  |
471  |
472  |
473  |
474  |
475  |
476  |
477  |
478  |
479  |
480  |
481  |
482  |
483  |
484  |
485  |
486  |
487  |
488  |
489  |
490  |
491  |
492  |
493  |
494  |
495  |
496  |
497  |
498  |
499  |
500  |
501  |
502  |
503  |
504  |
505  |
506  |
507  |
508  |
509  |
510  |
511  |
512  |
513  |
514  |
515  |
516  |
517  |
518  |
519  |
520  |
521  |
522  |
523  |
524  |
525  |
526  |
527  |
528  |
529  |
530  |
531  |
532  |
533  |
534  |
535  |
536  |
537  |
538  |
539  |
540  |
541  |
542  |
543  |
544  |
545  |
546  |
547  |
548  |
549  |
550  |
551  |
552  |
553  |
554  |
555  |
556  |
557  |
558  |
559  |
560  |
561  |
562  |
563  |
564  |
565  |
566  |
567  |
568  |
569  |
570  |
571  |
572  |
573  |
574  |
575  |
576  |
577  |
578  |
579  |
580  |
581  |
582  |
583  |
584  |
585  |
586  |
587  |
588  |
589  |
590  |
591  |
592  |
593  |
594  |
595  |
596  |
597  |
598  |
599  |
600  |
601  |
602  |
603  |
604  |
605  |
606  |
607  |
608  |
609  |
610  |
611  |
612  |
613  |
614  |
615  |
616  |
617  |
618  |
619  |
620  |
621  |
622  |
623  |
624  |
625  |
626  |
627  |
628  |
629  |
630  |
631  |
632  |
633  |
634  |
635  |
636  |
637  |
638  |
639  |
640  |
641  |
642  |
643  |
644  |
645  |
646  |
647  |
648  |
649  |
650  |
651  |
652  |
653  |
654  |
655  |
656  |
657  |
658  |
659  |
660  |
661  |
662  |
663  |
664  |
665  |
666  |
667  |
668  |
669  |
670  |
671  |
672  |
673  |
674  |
675  |
676  |
677  |
678  |
679  |
680  |
681  |
682  |
683  |
684  |
685  |
686  |
687  |
688  |
689  |
690  |
691  |
692  |
693  |
694  |
695  |
696  |
697  |
698  |
699  |
700  |
701  |
702  |
703  |
704  |
705  |
706  |
707  |
708  |
709  |
710  |
711  |
712  |
713  |
714  |
715  |
716  |
717  |
718  |
719  |
720  |
721  |
722  |
723  |
724  |
725  |
726  |
727  |
728  |
729  |
730  |
731  |
732  |
733  |
734  |
735  |
736  |
737  |
738  |
739  |
740  |
741  |
742  |
743  |
744  |
745  |
746  |
747  |
748  |
749  |
750  |
751  |
752  |
753  |
754  |
755  |
756  |
757  |
758  |
759  |
760  |
761  |
762  |
763  |
764  |
765  |
766  |
767  |
768  |
769  |
770  |
771  |
772  |
773  |
774  |
775  |
776  |
777  |
778  |
779  |
780  |
781  |
782  |
783  |
784  |
785  |
786  |
787  |
788  |
789  |
790  |
791  |
792  |
793  |
794  |
795  |
796  |
797  |
798  |
799  |
800  |
801  |
802  |
803  |
804  |
805  |
806  |
807  |
808  |
809  |
810  |
811  |
812  |
813  |
814  |
815  |
816  |
817  |
818  |
819  |
820  |
821  |
822  |
823  |
824  |
825  |
826  |
827  |
828  |
829  |
830  |
831  |
832  |
833  |
834  |
835  |
836  |
837  |
838  |
839  |
840  |
841  |
842  |
843  |
844  |
845  |
846  |
847  |
848  |
849  |
850  |
851  |
852  |
853  |
854  |
855  |
856  |
857  |
858  |
859  |
860  |
861  |
862  |
863  |
864  |
865  |
866  |
867  |
868  |
869  |
870  |
871  |
872  |
873  |
874  |
875  |
876  |
877  |
878  |
879  |
880  |
881  |
882  |
883  |
884  |
885  |
886  |
887  |
888  |
889  |
890  |
891  |
892  |
893  |
894  |
895  |
896  |
897  |
898  |
899  |
900  |
901  |
902  |
903  |
904  |
905  |
906  |
907  |
908  |
909  |
910  |
911  |
912  |
913  |
914  |
915  |
916  |
917  |
918  |
919  |
920  |
921  |
922  |
923  |
924  |
925  |
926  |
927  |
928  |
929  |
930  |
931  |
932  |
933  |
934  |
935  |
936  |
937  |
938  |
939  |
940  |
941  |
942  |
943  |
944  |
945  |
946  |
947  |
948  |
949  |
950  |
951  |
952  |
953  |
954  |
955  |
956  |
957  |
958  |
959  |
960  |
961  |
962  |
963  |
964  |
965  |
966  |
967  |
968  |
969  |
970  |
971  |
972  |
973  |
974  |
975  |
976  |
977  |
978  |
979  |
980  |
981  |
982  |
983  |
984  |
985  |
986  |
987  |
988  |
989  |
990  |
991  |
992  |
993  |
994  |
995  |
996  |
997  |
998  |
999  |
1000 |
1001 |
1002 |
1003 |
1004 |
1005 |
1006 |
1007 |
1008 |
1009 |
1010 |
1011 |
1012 |
1013 |
1014 |
1015 |
1016 |
1017 |
1018 |
1019 |
1020 |
1021 |
1022 |
1023 |
1024 |
1025 |
1026 |
1027 |
1028 |
1029 |
1030 |
1031 |
1032 |
1033 |
1034 |
1035 |
1036 |
1037 |
1038 |
1039 |
1040 |
1041 |
1042 |
1043 |
1044 |
1045 |
1046 |
1047 |
1048 |
1049 |
1050 |
1051 |
1052 |
1053 |
1054 |
1055 |
1056 |
1057 |
1058 |
1059 |
1060 |
1061 |
1062 |
1063 |
1064 |
1065 |
1066 |
1067 |
1068 |
1069 |
1070 |
1071 |
1072 |
1073 |
1074 |
1075 |
1076 |
1077 |
1078 |
1079 |
1080 |
1081 |
1082 |
1083 |
1084 |
1085 |
1086 |
1087 |
1088 |
1089 |
1090 |
1091 |
1092 |
1093 |
1094 |
1095 |
1096 |
1097 |
1098 |
1099 |
1100 |
1101 |
1102 |
1103 |
1104 |
1105 |
1106 |
1107 |
1108 |
1109 |
1110 |
1111 |
1112 |
1113 |
1114 |
1115 |
1116 |
1117 |
1118 |
1119 |
1120 |
1121 |
1122 |
1123 |
1124 |
1125 |
1126 |
1127 |
1128 |
1129 |
1130 |
1131 |
1132 |
1133 |
1134 |
1135 |
1136 |
1137 |
1138 |
1139 |
1140 |
1141 |
1142 |
1143 |
1144 |
1145 |
1146 |
1147 |
1148 |
1149 |
1150 |
1151 |
1152 |
1153 |
1154 |
1155 |
1156 |
1157 |
1158 |
1159 |
1160 |
1161 |
1162 |
1163 |
1164 |
1165 |
1166 |
1167 |
1168 |
1169 |
1170 |
1171 |
1172 |
1173 |
1174 |
1175 |
1176 |
1177 |
1178 |
1179 |
1180 |
1181 |
1182 |
1183 |
1184 |
1185 |
1186 |
1187 |
1188 |
1189 |
1190 |
1191 |
1192 |
1193 |
1194 |
1195 |
1196 |
1197 |
1198 |
1199 |
1200 |
1201 |
1202 |
1203 |
1204 |
1205 |
1206 |
1207 |
1208 |
1209 |
1210 |
1211 |
1212 |
1213 |
1214 |
1215 |
1216 |
1217 |
1218 |
1219 |
1220 |
1221 |
1222 |
1223 |
1224 |
1225 |
1226 |
1227 |
1228 |
1229 |
1230 |
1231 |
1232 |
1233 |
1234 |
1235 |
1236 |
1237 |
1238 |
1239 |
1240 |
1241 |
1242 |
1243 |
1244 |
1245 |
1246 |
1247 |
1248 |
1249 |
1250 |
1251 |
1252 |
1253 |
1254 |
1255 |
1256 |
1257 |
1258 |
1259 |
1260 |
1261 |
1262 |
1263 |
1264 |
1265 |
1266 |
1267 |
1268 |
1269 |
1270 |
1271 |
1272 |
1273 |
1274 |
1275 |
1276 |
1277 |
1278 |
1279 |
1280 |
1281 |
1282 |
1283 |
1284 |
1285 |
1286 |
1287 |
1288 |
1289 |
1290 |
1291 |
1292 |
1293 |
1294 |
1295 |
1296 |
1297 |
1298 |
1299 |
1300 |
1301 |
1302 |
1303 |
1304 |
1305 |
1306 |
1307 |
1308 |
1309 |
1310 |
1311 |
1312 |
1313 |
1314 |
1315 |
1316 |
1317 |
1318 |
1319 |
1320 |
1321 |
1322 |
1323 |
1324 |
1325 |
1326 |
1327 |
1328 |
1329 |
1330 |
1331 |
1332 |
1333 |
1334 |
1335 |
1336 |
1337 |
1338 |
1339 |
1340 |
1341 |
1342 |
1343 |
1344 |
1345 |
1346 |
1347 |
1348 |
1349 |
1350 |
1351 |
1352 |
1353 |
1354 |
1355 |
1356 |
1357 |
1358 |
1359 |
1360 |
1361 |
1362 |
1363 |
1364 |
1365 |
1366 |
1367 |
1368 |
1369 |
1370 |
1371 |
1372 |
1373 |
1374 |
1375 |
1376 |
1377 |
1378 |
1379 |
1380 |
1381 |
1382 |
1383 |
1384 |
1385 |
1386 |
1387 |
1388 |
1389 |
1390 |
1391 |
1392 |
1393 |
1394 |
1395 |
1396 |
1397 |
1398 |
1399 |
1400 |
1401 |
1402 |
1403 |
1404 |
1405 |
1406 |
1407 |
1408 |
1409 |
1410 |
1411 |
1412 |
1413 |
1414 |
1415 |
1416 |
1417 |
1418 |
1419 |
1420 |
1421 |
1422 |
1423 |
1424 |
1425 |
1426 |
1427 |
1428 |
1429 |
1430 |
1431 |
1432 |
1433 |
1434 |
1435 |
1436 |
1437 |
1438 |
1439 |
1440 |
1441 |
1442 |
1443 |
1444 |
1445 |
1446 |
1447 |
1448 |
1449 |
1450 |
1451 |
1452 |
1453 |
1454 |
1455 |
1456 |
1457 |
1458 |
1459 |
1460 |
1461 |
1462 |
1463 |
1464 |
1465 |
1466 |
1467 |
1468 |
1469 |
1470 |
1471 |
1472 |
1473 |
1474 |
1475 |
1476 |
1477 |
1478 |
1479 |
1480 |
1481 |
1482 |
1483 |
1484 |
1485 |
1486 |
1487 |
1488 |
1489 |
1490 |
1491 |
1492 |
1493 |
1494 |
1495 |
1496 |
1497 |
1498 |
1499 |
1500 |
1501 |
1502 |
1503 |
1504 |
1505 |
1506 |
1507 |
1508 |
1509 |
1510 |
1511 |
1512 |
1513 |
1514 |
1515 |
1516 |
1517 |
1518 |
1519 |
1520 |
1521 |
1522 |
1523 |
1524 |
1525 |
1526 |
1527 |
1528 |
1529 |
1530 |
1531 |
1532 |
1533 |
1534 |
1535 |
1536 |
1537 |
1538 |
1539 |
1540 |
1541 |
1542 |
1543 |
1544 |
1545 |
1546 |
1547 |
1548 |
1549 |
1550 |
1551 |
1552 |
1553 |
1554 |
1555 |
1556 |
1557 |
1558 |
1559 |
1560 |
1561 |
1562 |
1563 |
1564 |
1565 |
1566 |
1567 |
1568 |
1569 |
1570 |
1571 |
1572 |
1573 |
1574 |
1575 |
1576 |
1577 |
1578 |
1579 |
1580 |
1581 |
1582 |
1583 |
1584 |
1585 |
1586 |
1587 |
1588 |
1589 |
1590 |
1591 |
1592 |
1593 |
1594 |
1595 |
1596 |
1597 |
1598 |
1599 |
1600 |
1601 |
1602 |
1603 |
1604 |
1605 |
1606 |
1607 |
1608 |
1609 |
1610 |
1611 |
1612 |
1613 |
1614 |
1615 |
1616 |
1617 |
1618 |
1619 |
1620 |
1621 |
1622 |
1623 |
1624 |
1625 |
1626 |
1627 |
1628 |
1629 |
1630 |
1631 |
1632 |
1633 |
1634 |
1635 |
1636 |
1637 |
1638 |
1639 |
1640 |
1641 |
1642 |
1643 |
1644 |
1645 |
1646 |
1647 |
1648 |
1649 |
1650 |
1651 |
1652 |
1653 |
1654 |
1655 |
1656 |
1657 |
1658 |
1659 |
1660 |
1661 |
1662 |
1663 |
1664 |
1665 |
1666 |
1667 |
1668 |
1669 |
1670 |
1671 |
1672 |
1673 |
1674 |
1675 |
1676 |
1677 |
1678 |
1679 |
1680 |
1681 |
1682 |
1683 |
1684 |
1685 |
1686 |
1687 |
1688 |
1689 |
1690 |
1691 |
1692 |
1693 |
1694 |
1695 |
1696 |
1697 |
1698 |
1699 |
1700 |
1701 |
1702 |
1703 |
1704 |
1705 |
1706 |
1707 |
1708 |
1709 |
1710 |
1711 |
1712 |
1713 |
1714 |
1715 |
1716 |
1717 |
1718 |
1719 |
1720 |
1721 |
1722 |
1723 |
1724 |
1725 |
1726 |
1727 |
1728 |
1729 |
1730 |
1731 |
1732 |
1733 |
1734 |
1735 |
1736 |
1737 |
1738 |
1739 |
1740 |
1741 |
1742 |
1743 |
1744 |
1745 |
1746 |
1747 |
1748 |
1749 |
1750 |
1751 |
1752 |
1753 |
1754 |
1755 |
1756 |
1757 |
1758 |
1759 |
1760 |
1761 |
1762 |
1763 |
1764 |
1765 |
1766 |
1767 |
1768 |
1769 |
1770 |
1771 |
1772 |
1773 |
1774 |
1775 |
1776 |
1777 |
1778 |
1779 |
1780 |
1781 |
1782 |
1783 |
1784 |
1785 |
1786 |
1787 |
1788 |
1789 |
1790 |
1791 |
1792 |
1793 |
1794 |
1795 |
1796 |
1797 |
1798 |
1799 |
1800 |
1801 |
1802 |
1803 |
1804 |
1805 |
1806 |
1807 |
1808 |
1809 |
1810 |
1811 |
1812 |
1813 |
1814 |
1815 |
1816 |
1817 |
1818 |
1819 |
1820 |
1821 |
1822 |
1823 |
1824 |
1825 |
1826 |
1827 |
1828 |
1829 |
1830 |
1831 |
1832 |
1833 |
1834 |
1835 |
1836 |
1837 |
1838 |
1839 |
1840 |
1841 |
1842 |
1843 |
1844 |
1845 |
1846 |
1847 |
1848 |
1849 |
1850 |
1851 |
1852 |
1853 |
1854 |
1855 |
1856 |
1857 |
1858 |
1859 |
1860 |
1861 |
1862 |
1863 |
1864 |
1865 |
1866 |
1867 |
1868 |
1869 |
1870 |
1871 |
1872 |
1873 |
1874 |
1875 |
1876 |
1877 |
1878 |
1879 |
1880 |
1881 |
1882 |
1883 |
1884 |
1885 |
1886 |
1887 |
1888 |
1889 |
1890 |
1891 |
1892 |
1893 |
1894 |
1895 |
1896 |
1897 |
1898 |
1899 |
1900 |
1901 |
1902 |
1903 |
1904 |
1905 |
1906 |
1907 |
1908 |
1909 |
1910 |
1911 |
1912 |
1913 |
1914 |
1915 |
1916 |
1917
```

é feita uma filtragem dos nós que poderão ser eleitos como nós MPR, enquanto que na segunda fase é executado o processo de eleição dos nós MPR. A heurística existente no processo de eleição de nós MPR do protocolo OLSR, de que devem ser eleitos tantos nós MPR, quantos os necessários para garantir a total cobertura da vizinhança a 2-saltos, é desprezada. Na verdade, a selecção exclusiva de nós estáveis como nós MPR não garante a cobertura total da rede, pois tal necessidade origina um aumento do número de nós MPR, e consequente aumento do número de difusões de mensagens TC. A nova abordagem, permite uma redução do tráfego de *broadcast* gerado pelos nós MPR na difusão de mensagens TC, e consequentemente uma diminuição da utilização dos recursos de largura de banda utilizados, diminuindo a carga na rede relativa à difusão das mensagens TC.

De modo a adaptar o algoritmo de agrupamento de nós ao protocolo de encaminhamento OLSR, é necessário que seja associado a cada nó uma tabela de *beacons* (Tabela 4.2). Na verdade, e tendo em conta que o protocolo OLSR associa a cada nó uma tabela de vizinhança, toda a informação existente na tabela de *beacons* pode ser integrada na tabela de vizinhança. É também necessário modificar a estrutura da mensagem HELLO, sendo:

- Adicionado o campo *is_bgl* de modo a identificar se o nó é LGB, necessário para o método de eleição de nós MPR.
- Adicionado o campo *gl_address* que contém o endereço do nó LGB eleito pelo emissor da mensagem HELLO. Este campo é necessário para o algoritmo de agrupamento de nós.
- Modificada a utilização do campo *willingness*. Deixa de referir a "vontade" de um nó em ser um nó MPR, que se baseava no estado de energia do aparelho, embora continue a ser utilizada como uma variável de controlo de grande importância na eleição de nós MPR. Como referido na Secção 2.1, em redes veiculares a limitação de energia existente no dispositivo é corrigida com a utilização das baterias dos veículos, podendo então ser alterada a funcionalidade desta variável. Nos parágrafos seguintes é explicado com mais detalhe a sua nova utilização.

Com a adaptação do algoritmo de agrupamento de nós, o método de eleição de nós MPR

tem início muito tempo antes de se chegar ao procedimento de eleição de nós MPR, onde por norma, no protocolo OLSR é feita a eleição. No envio das mensagens HELLO, os campos *gl_address* e *is_bgl* são retirados da tabela de *beacons* associada ao nó emissor, e adicionados à mensagem HELLO. O campo *willingness* é utilizado no processo de filtragem do conjunto de nós que podem ser eleitos como nós MPR. O nó emissor da mensagem HELLO verifica, através da sua tabela de vizinhança, se é considerado por qualquer nó, um nó estável, ou seja, se tem alguma ligação com uma duração superior ao limiar de estabilidade (k_{est}). Caso seja considerado um nó estável, coloca o valor *WILL_DEFAULT* na variável *willingness* indicando a sua estabilidade, caso contrário coloca o valor *WILL_NEVER*, demonstrando a sua instabilidade.

Quando a mensagem HELLO é recebida é necessário fazer a actualização da tabela de *beacons*, e também dar continuação ao processo de filtragem de nós que poderão ser eleitos nós MPR, iniciado no envio da mensagem HELLO. Depois de actualizada a tabela de *beacons*, são guardados todos os valores de estabilidade (η), referentes às ligações que o nó receptor da mensagem HELLO tem para com todos os seus vizinhos (n_y). Esta informação será utilizada mais à frente, no procedimento de eleição de nós MPR propriamente dito.

A informação de estabilidade relativa ao nó emissor, que foi enviada através da variável *willingness* é agora actualizada, tendo em conta a relação existente entre o nó emissor da mensagem HELLO e o nó receptor. Todos os nós, que no envio da mensagem HELLO eram considerados como nós instáveis (*willingness* = *WILL_NEVER*), assim permanecerão. Esta filtragem, aplicada aos cenários de auto-estrada, vai permitir que qualquer nó que não tenha uma ligação estável com qualquer outro nó, seja do mesmo sentido ou do sentido contrário, nunca possa ser eleito um nó MPR. No que diz respeito aos restantes nós, aqueles que enviaram *willingness* = *WILL_DEFAULT*, é necessário actualizar a variável *willingness*, verificando agora a relação de estabilidade relativamente ao nó que recebeu a mensagem HELLO:

- **Regra 1:** Se o nó que enviou a mensagem é um nó LGB, e se apresentar uma ligação com o nó receptor com uma duração igual ou superior ao limiar de estabilidade k_{est}

(ser um nó estável segundo a Definição 4.6), então a variável *willingness* é alterada para *WILL_ALWAYS*. Aplicando aos cenários da auto-estrada, esta regra contempla todos os nós LGB que viajam no mesmo sentido do nó receptor da mensagem HELLO.

- **Regra 2:** Se o nó que enviou a mensagem não for LGB mas apresentar uma ligação com o nó receptor com uma duração igual ou superior ao limiar de estabilidade k_{est} , então a variável *willingness* mantém-se em *WILL_DEFAULT*. Esta regra contempla todos os nós que viajam no mesmo sentido do nó receptor da mensagem HELLO, mas que não foram eleitos LGB.
- **Regra 3:** Os restantes nós que não são considerados nas regras anteriores, vêm a sua *willingness* alterada para *WILL_NEVER*. São todos os nós que conseguiram entrar neste processo de filtragem por serem considerados estáveis por qualquer nó da rede, mas que não apresentam uma relação estável com o nó receptor. Ou seja, são todos os nós que circulam em sentido contrário.

Com o processo de filtragem e o conjunto de regras apresentado anteriormente, é possível agora por parte do protocolo OLSR, perceber quais são os nós que viajam no mesmo sentido. Contudo, como a funcionalidade da variável *willingness* foi modificada, o tradicional processo de eleição de nós MPR, do protocolo OLSR, tem também de ser modificado. No entanto, as alterações propostas ao processo de eleição de nós MPR não se devem apenas a este facto. Como já foi referido anteriormente, a garantia de uma cobertura total dos vizinhos a 2-saltos, por parte dos nós MPR é desprezada, e a regra R3 do Algoritmo 5.1 terá de ser retirada.

Assim, o remodelado processo de eleição do conjunto de nós MPR do nó genérico n_a , apresentado no Algoritmo 5.3, continua a manter os como parâmetros de entrada as vizinhanças N_a e N_{2a} , existindo agora outro parâmetro designado por "cobertura_desejada". Este parâmetro é utilizado para controlar a quantidade de nós MPR eleitos, em função da cobertura da vizinhança a 2-saltos. É importante relembrar que, segundo a Definição 5.4, os nós vizinhos a 2-saltos, que apenas são alcançáveis por nós vizinhos a 1-salto com

```

parâmetros de entrada      :  $N_a, N_{2a}, \eta(n_y) (\forall n_y \in N_a), cobertura\_desejada$ 
parâmetros de saída       :  $MPR(n_a)$ 

1  cobertura  $\leftarrow$  0
2  for cada nó vizinho a 1-salto ( $n_y \in N_a$ ) do
3      if  $willingness(n_y) = WILL\_ALWAYS$  then                                /* Regra NR1 */
4          insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
5          remove_nós( $N_{2a}, n_y$ )
6
7  cobertura = calcula_cobertura( $N_{2a}$ )
8  while  $cobertura < cobertura\_desejada$  do                                /* Regra NR3 */
9      calcular_alcancabilidade( $n_a, n_y$ )                                /* Regra NR3.1 */
10     if  $\eta(n_y) > \eta(n), n \in N_a$  then                                /* Regra NR3.2 */
11         insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
12     else
13         if  $\eta(n_y) = \eta(n), n \in N_a$  then
14             if  $alcancabilidade(n_y, n_a) > alcancabilidade(n, n_a), n \in N_a$  then
15                 insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
16             else
17                 if  $alcancabilidade(n_y, n_a) = alcancabilidade(n, n_a), n \in N_a$  then
18                     if  $degree(n_y) > degree(n), n \in N_a$  then                /* Regra NR2 */
19                         insere_no_conjunto_de_nós_MPR( $n_a, n_y$ )
20
21
22
23
24     remove_nós( $N_{2a}, n_y$ )
25     cobertura = calcula_cobertura( $N_{2a}$ )
26

```

Algoritmo 5.3: Algoritmo de eleição do conjunto de nós MPR do nó genérico n_a , modificado de forma a contemplar a alterações propostas.

$willingness = WILL_NEVER$ não fazem parte do conjunto N_{2a} . Desta forma, todos nós que circulam no sentido contrário, e que foram rotulados no processo de filtragem apresentado anteriormente com $willingness = WILL_NEVER$, nunca poderão ser eleitos como nós MPR. São propostas as seguintes regras (designadas por Novas Regras (NR)):

- **NR 1:** Todos os nós vizinhos a 1-salto de n_a que apresentem $willingness = WILL_ALWAYS$, que identificam veículos LGB que circulam no mesmo sentido de n_a , são automaticamente eleitos para o conjunto de nós MPR (linhas 3 a 6).
- **NR 2:** Calcula-se $degree(n_y)$ para todos os nós vizinhos a 1-salto de n_a (linha 19). Apesar de ser a regra NR2, só será aplicada durante a execução da regra NR 3.2.
- **NR 3:** Enquanto o conjunto de nós MPR de n_a não cobrir o número de nós n_{2y} suficiente, de modo a que a cobertura desejada tenha sido alcançada:
 - **NR 3.1:** Para cada nó vizinho n_y calcular a sua alcancabilidade (linha 10).

- **NR 3.2:** Elege-se como nó MPR o nó n_y que apresentar maior valor de estabilidade (η) de entre todos os nós N_a que não apresentarem uma alcançabilidade nula. Com esta modificação garante-se que os nós que circulam no mesmo sentido, mas que não são LGB, são eleitos segundo uma ordem descendente da duração da ligação que têm com o nó n_a . Ou seja, os nós que apresentem uma ligação mais duradoura com n_a , e consequentemente mais estável, têm prioridade em ser nós MPR de n_a . Em caso de empate deve-se eleger o nó que fornecer maior alcançabilidade. Novamente em caso de empate, é eleito o nó que apresentar um maior valor de *degree* (linhas 11 a 20).

À medida que as regras vão sendo aplicadas, são retirados do subconjunto N_{2a} , os nós vizinhos a 2-saltos que vão estando cobertos pelos nós vizinhos a 1-salto, eleitos com nós MPR.

Com as propostas anteriormente apresentadas, o protocolo de encaminhamento OLSR é agora capaz de fazer uma eleição do conjunto de nós MPR baseado na estabilidade das ligações. Para que tal seja possível, as heurísticas existentes no método de eleição de nós MPR do OLSR, que se baseavam no estado de energia de cada nó, e que garantiam uma cobertura total de vizinhos a 2-saltos, foram eliminadas.

Durante este capítulo, estudou-se o modo de funcionamento do protocolo de encaminhamento com optimização de topologia OLSR. Foram descritos os métodos de eleição de nós MPR e de elaboração da tabela de encaminhamento, sendo também apresentadas as alterações propostas ao método de eleição de nós MPR. Este conjunto de alterações são focadas no conjunto de heurísticas que regem o método de eleição do conjunto de nós MPR, que visam melhorar o desempenho do protocolo OLSR, em cenários de redes veiculares com a existência de tráfego em ambos os sentidos. No capítulo seguinte é feita a avaliação de desempenho do protocolo OLSR modificado.

Capítulo 6

Análise de Desempenho

Uma vez apresentadas as modificações propostas ao protocolo de encaminhamento OLSR, são apresentados neste capítulo os testes realizados, assim como os resultados obtidos, por forma a comparar o desempenho entre o protocolo de encaminhamento OLSR e o protocolo OLSR contendo as alterações propostas. As métricas utilizadas na comparação dos protocolos baseiam-se sobretudo na taxa de sucesso de resolução dos pedidos de encaminhamento, bem como no tempo necessário à sua resolução (atraso do caminho).

Este capítulo encontra-se dividido em três secções. Na primeira secção é realizada uma descrição da simulação, com a indicação dos vários parâmetros utilizados, e das métricas que foram utilizadas para avaliar o desempenho dos dois protocolos. Na segunda secção é apresentado o método utilizado para tratar os resultados obtidos nas simulações. Por fim, na terceira e última secção são apresentados os resultados obtidos nos testes realizados.

6.1 Descrição da Simulação

Para simular o protocolo OLSR foi utilizada a realização do protocolo disponibilizado em [Ros09], para o simulador de redes *ns-2*. O código disponibilizado em [Ros09] serviu também de base à integração do algoritmo de agrupamento de nós, tendo sido alterado o método de eleição do conjunto de nós MPR, como referido na Secção 5.2. Esta modificação ao protocolo de encaminhamento OLSR será referenciada de ora em diante como

OLSR-FCT¹. Foi utilizada a realização da norma IEEE 802.11 [Sta05], disponibilizada no simulador *ns-2*, para operar a nível físico e a sub-nível MAC.

Na norma IEEE 802.11 foi utilizada a parametrização por omissão do simulador, embora tenham sido alterados o alcance de rádio para 1000 *m*, e o tempo de expiração de *acknowledge*, por forma a garantir a transmissão das mensagens de *acknowledge* dentro do intervalo de *timeout*, referente ao funcionamento desta norma. Os ritmos de transmissão utilizados no envio de tráfego *broadcast* (mensagens HELLO) foram de 2 *Mbps*, e 11 *Mbps* para tráfego *unicast*.

O desempenho dos protocolos OLSR e OLSR-FCT foi avaliado através da geração de pedidos de encaminhamento. Foi desenvolvida uma aplicação de tráfego, que estando associada a cada nó, é responsável pelo envio de pacotes que originam os pedidos de encaminhamento. Definiu-se uma periodicidade de pedido de encaminhamento de 2 segundos para cada nó, sempre que este se encontre activo, ou seja, sempre que o nó se encontra a circular na auto-estrada. O destino do pedido de encaminhamento é escolhido aleatoriamente de entre o nós que no momento se encontrarem em movimento e apenas para os veículos que circulam na mesma faixa. Na Tabela 6.1 é apresentado o número médio de pedidos de encaminhamento realizados durante uma simulação, para os 4 cenários de densidade.

Tabela 6.1: Número médio total de pedidos de encaminhamento realizados durante uma simulação, nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

	Cen_{4v}	Cen_{6v}	Cen_{8v}	Cen_{10v}
Número Médio Total de Pedidos de Encaminhamento por Simulação	5406	7388	10102	13220

Os pedidos de encaminhamento são registados, obtendo-se estatísticas de percentagem de sucesso de resolução dos pedidos, bem como do tempo necessário à sua resolução (também designado por "atraso do caminho"), sendo estas as métricas utilizadas para comparar o desempenho dos protocolos OLSR e OLSR-FCT. Existem várias razões para que o pe-

¹A sigla FCT refere-se à Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, instituição onde esta dissertação foi desenvolvida.

dido de encaminhamento falhe, as quais podem estar relacionadas com a inexistência de caminho devido à existência de grupos de nós desconectados, a estados de incoerência do protocolo de encaminhamento, a um elevado nível de tráfego na rede (congestão), ou até mesmo à existência de ciclos que originam perda de pacotes por ter excedido o valor de *time-to-live* (TTL). Através da escolha de MPR mais estáveis, este trabalho pretende diminuir o número de pedidos de encaminhamento falhados, devido à incoerência do protocolo e da diminuição de situações de congestão observadas na rede.

Relativamente aos protocolos OLSR e OLSR-FCT parametrizou-se o período de envio de mensagens HELLO com 1 *segundo*, enquanto que as mensagens de topologia (TC) são enviadas com uma periodicidade de 2 *segundos*. No algoritmo de controlo de topologia, como o *beacon* foi incorporado na mensagem HELLO do protocolo OLSR, o período de *beacon* (T_B) é de 1 *segundo*. O parâmetro T_O associado ao *timeout* do *beacon* tem uma parametrização diferente para cada cenário. No final do Capítulo 4 observou-se a influência da parametrização do T_O no conhecimento da topologia da rede, e como tal, para cada cenário de simulação é utilizado o valor de T_O que possa garantir um melhor conhecimento da topologia da rede (Tabela 6.2). No entanto, durante este capítulo é estudada a influência deste parâmetro no desempenho do protocolo OLSR-FCT.

Tabela 6.2: Valores de T_O nos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

	Cen_{4v}	Cen_{6v}	Cen_{8v}	Cen_{10v}
T_O (s)	2.5	3.5	4.5	5.5

O limiar de estabilidade k_{est} foi parametrizado com o valor de 50, que como já foi referido anteriormente, possibilita a distinção entre ligações de nós que viajam no mesmo sentido, e ligações entre nós que viajam em sentidos opostos. Contudo é apresentada, para o cenário Cen_{6v} , a influência da parametrização do limiar de estabilidade k_{est} no desempenho do OLSR-FCT.

Como descrito na Secção 4.3, o conjunto de nós eleitos como nós MPR, no protocolo OLSR-FCT, pode não garantir a cobertura total da rede. Ou seja, poderão existir nós

vizinhos a 2-saltos do nó n que não estão cobertos pelo conjunto de nós MPR do nó n . Como tal, para cada cenário de simulação, foram definidos quatro valores de cobertura, com o intuito de avaliar o desempenho das modificações propostas, em função da cobertura desejada: 45%, 60%, 85% e 100%. Por sua vez, o funcionamento do protocolo OLSR tenta, sempre que possível, garantir uma cobertura total do conjunto de vizinhos a 2-saltos.

6.2 Método de Recolha de Dados

Por forma a garantir um intervalo de confiança a 95% dos resultados apresentados na secção seguinte, foram realizadas, para cada cenário e para cada valor de cobertura desejada, 10 simulações. Em cada simulação, a semente do gerador de números aleatórios da aplicação de tráfego é modificada, garantindo que em cada simulação os nós de destinos do pedido de encaminhamento sejam diferentes.

O método utilizado para medir o desempenho dos dois protocolos de encaminhamento, para os diferentes cenários de simulação e diferentes valores de cobertura, foi o método das réplicas [Tah87], que refere que deve ser reunido um conjunto de amostras independentes, realizando posteriormente um tratamento estatísticos das mesmas. A partir do conjunto de amostras (x_i) é possível obter o seu valor médio (\bar{X}) e a sua variância (s^2) . Com estes dados é possível determinar o seu intervalo de confiança. Quando a variável aleatória tem uma distribuição normal, o valor esperado da variável aleatória está compreendido no intervalo obtido pela Equação (6.1), com uma probabilidade $1 - \varphi$ e um número de amostras n , onde o parâmetro $t_{\frac{\varphi}{2}}$ é obtido através da consulta da distribuição t de *Students*.

$$(\bar{X}) - t_{\frac{\varphi}{2}} \cdot \frac{s}{\sqrt{n}} \leq \mu \leq (\bar{X}) + t_{\frac{\varphi}{2}} \cdot \frac{s}{\sqrt{n}} \quad (6.1)$$

Nesta dissertação foi utilizado $\varphi = 5\%$, o que garante uma margem de erro a 95% de confiança.

6.3 Resultados

Na Secção 1.2 observou-se a degradação do desempenho do protocolo de encaminhamento OLSR, com a existência de tráfego nos 2 sentidos. Antes de se proceder à comparação entre o protocolo OLSR e o OLSR-FCT, começa-se por analisar o comportamento do OLSR face à existência de diferentes densidades de veículos. Na Figura 6.1(a) é apresentada a percentagem de sucesso dos pedidos de encaminhamento, para os quatro cenários de simulação. À medida que a densidade vai aumentando a percentagem de sucesso sofre uma ligeira melhoria, contudo a partir do cenário de densidade de 8 vizinhos o sucesso começa a diminuir.

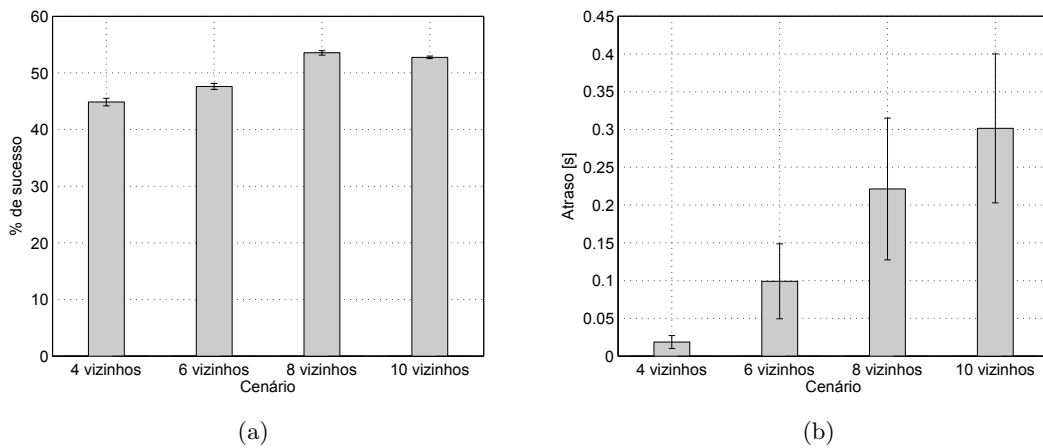


Figura 6.1: Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), do protocolo OLSR, aplicado aos cenários Cen_{4v} , Cen_{6v} , Cen_{8v} e Cen_{10v} .

No entanto, como se pode observar pela Figura 6.1(b), o atraso médio dos caminhos vai aumentando para cenários de maior densidade, o que poderá evidenciar uma degradação do protocolo OLSR sempre que a densidade do cenário aumentar. Este facto é justificado pelo aumento do tráfego, devido ao aumento da densidade de veículos. O número de pacotes HELLO enviados está directamente relacionado com o número de veículos existentes, que como são transmitidos em modo *broadcast*, apresentam um ritmo de transmissão inferior ao modo *unicast*, ocupando o canal durante um período mais longo. Com a eleição de mais nós MPR, a geração e propagação de mensagens TC também aumenta, aumentando consequentemente a largura de banda consumida. O aumento do atraso pode ser ainda

justificado pela eleição de nós que viajam em sentido contrário como nós MPR, provocando situações de incoerência.

A Figura 6.2 apresenta a percentagem de sucesso dos pedidos de encaminhamento, e o atraso do caminho, no cenário Cen_{4v} , para os protocolos OLSR e OLSR-FCT. Pela observação da Figura 6.2(a), as modificações introduzidas ao protocolo OLSR degradam o seu desempenho. Este comportamento pode ser justificado pela existência de partições na rede, quando são utilizados apenas os veículos que viajam no mesmo sentido. Como o protocolo OLSR não diferencia os veículos que circulam no mesmo sentido, dos veículos que viajam no sentido oposto, as possíveis partições na rede são compensadas pela eleição de veículos que viajam no sentido oposto como nós MPR. Como se trata de um cenário de baixa densidade, não é possível observar a influência da cobertura no desempenho do protocolo OLSR-FCT, pois os resultados permanecem idênticos.

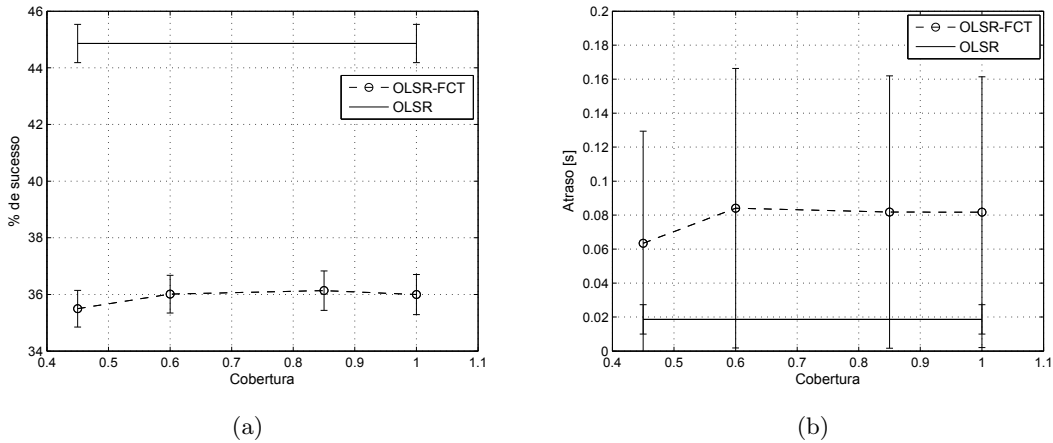


Figura 6.2: Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolos OLSR e OLSR-FCT, aplicado ao cenário Cen_{4v} , para diferentes valores de cobertura.

Relativamente ao atraso do caminho, o desempenho do protocolo OLSR continua a ser superior ao OLSR-FCT (Figura 6.2(b)). O intervalo de confiança é largo devido ao elevado desvio padrão das medidas. Este facto é explicado pelo diferente comprimento dos caminhos, pois o atraso do caminho é fortemente influenciado pelo número de nós que o constituem. Este facto não se verifica apenas neste cenário de densidade, mas também nos restantes, como se poderá observar nas figuras seguintes. Segundo os resultados obti-

dos pode-se afirmar que as modificações propostas ao OLSR-FCT não oferecem melhorias relativamente ao OLSR, em cenários de baixa densidade.

O cenário de 6 vizinhos (Cen_{6v}) apresenta resultados diferentes dos obtidos no cenário anterior. Como se pode ver pela Figura 6.3(a), a percentagem de sucesso de resoluções dos pedidos de encaminhamento no protocolo OLSR-FCT é superior ao do protocolo OLSR, em todos os cenários de cobertura. As melhorias representam um acréscimo relativo da percentagem de sucesso que vai desde os 13%, com uma cobertura de 45%, até 34% com a cobertura total da rede. Como existe neste cenário um aumento de densidade de veículos, relativamente ao cenário anterior, existem veículos suficientes no mesmo sentido capazes de criar um núcleo de nós que diminui a probabilidade da existência de uma rede particionada. Assim, o facto de só serem eleitos como veículos MPR aqueles que circulam no mesmo sentido, as ligações que são criadas entre os veículos e os seus veículos MPR, assim como entre veículos MPR, são mais estáveis, diminuindo as situações de incoerência. De notar ainda que à medida que a cobertura vai aumentando, a percentagem de sucesso também aumenta. Como a situação de maior cobertura significa maior quantidade de tráfego broadcast, é de realçar que o algoritmo permite traduzir essa situação numa melhoria de desempenho, pois o aumento de tráfego *broadcast* não influencia o desempenho do protocolo OLSR-FCT.

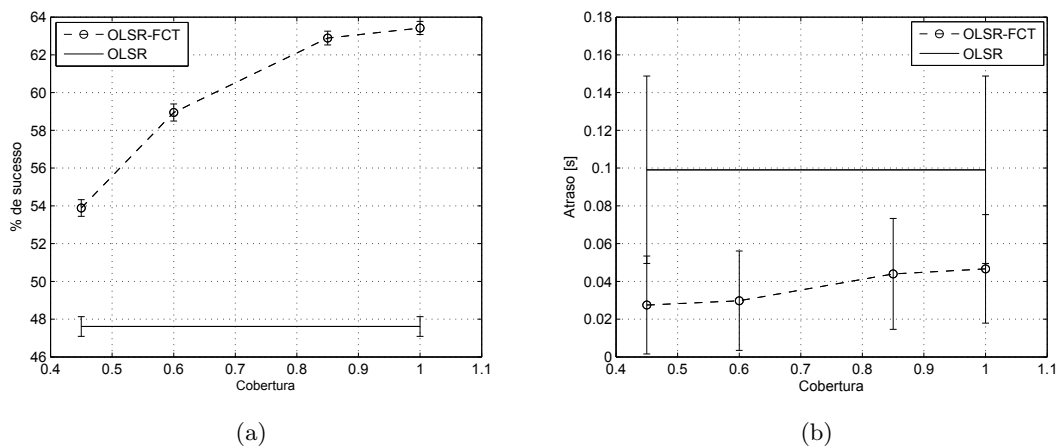


Figura 6.3: Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolos OLSR e OLSR-FCT, aplicado ao cenário Cen_{6v} , para diferentes valores de cobertura.

No que diz respeito ao atraso médio dos caminhos, também se verificam melhorias do protocolo OLSR-FCT relativamente ao seu originário. Pela Figura 6.3(b) é possível observar uma decréscimo relativo de aproximadamente 70%, no caso da cobertura mais baixa analisada, e um decréscimo relativo de 50% para o cenário de cobertura total. À medida que a cobertura vai aumentando, a melhoria no atraso médio dos caminhos é menos acentuada, justificada pelo aumento do número de veículos MPR eleitos, que gera mais tráfego *broadcast*, e consequente maior ocupação do canal. Estes resultados demonstram que os veículos eleitos como veículos MPR têm uma maior probabilidade de representarem caminhos com menor atraso (que podem ser inclusivamente os mais curtos). Assim, pode-se afirmar que para cenários de densidade de aproximadamente 6 vizinhos por sentido, o protocolo OLSR-FCT apresenta um desempenho significativamente superior ao protocolo OLSR.

A Figura 6.4 apresenta o desempenho do protocolo OLSR-FCT face ao OLSR no cenário de densidade média de 8 vizinhos. O comportamento do protocolo, em termos de percentagem de sucesso dos pedidos de encaminhamento, é idêntico ao cenário anterior (Cen_{6v}): para qualquer cobertura de rede, o protocolo OLSR-FCT apresenta melhores resultados do que o protocolo OLSR, melhorias estas que vão desde o aumento relativo de 9% para o cenário de menor densidade, até 26% para o cenário de cobertura total (Figura 6.4(a)).

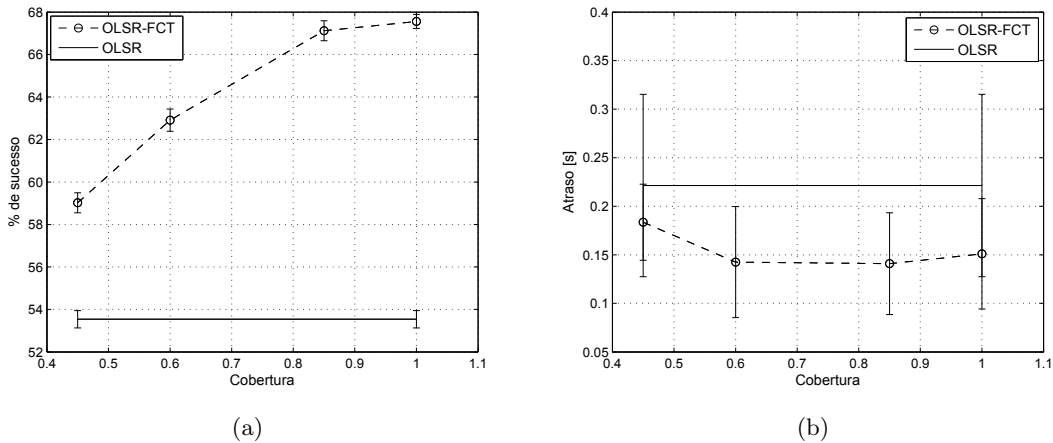


Figura 6.4: Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolos OLSR e OLSR-FCT, aplicado ao cenário Cen_{8v} , para diferentes valores de cobertura.

Relativamente ao atraso médio dos caminhos, o desempenho do OLSR-FCT é também superior ao protocolo OLSR. Pela observação da Figura 6.4(b) nota-se uma decréscimo relativo no tempo médio de atraso dos caminhos de 25% no cenário de menor cobertura, enquanto que a maior melhoria está nos cenários de 60% e 85% de cobertura, com um decréscimo relativo de aproximadamente 38%. No cenário de cobertura total, o desempenho do OLSR-FCT, relativamente ao atraso médio dos caminhos, diminui para os valores de menor cobertura, embora seja ainda melhor 33% do que o OLSR. Tal facto deve-se à necessidade de se elegerem mais veículos MPR para garantir a cobertura total da rede, e como tal, um conseqüente aumento da ocupação do canal, devido ao aumento do envio de mensagens TC.

No cenário de densidade média de 10 vizinhos (Cen_{10v}), o desempenho do protocolo OLSR-FCT face ao protocolo OLSR é idêntico aos dos dois cenários estudados anteriormente. Em termos de percentagem de sucesso de resolução de pedidos de encaminhamento, observa-se uma melhoria relativa de aproximadamente 11% para o cenário de cobertura de 45%, que atinge o seu melhor valor para o cenário de cobertura total, com um acréscimo relativo de 38% na percentagem de sucesso, como se pode observar pela Figura 6.5(a). O desempenho relativamente ao atraso médio dos caminhos apresenta um comportamento muito similar ao do cenário anterior (Cen_{8v}).

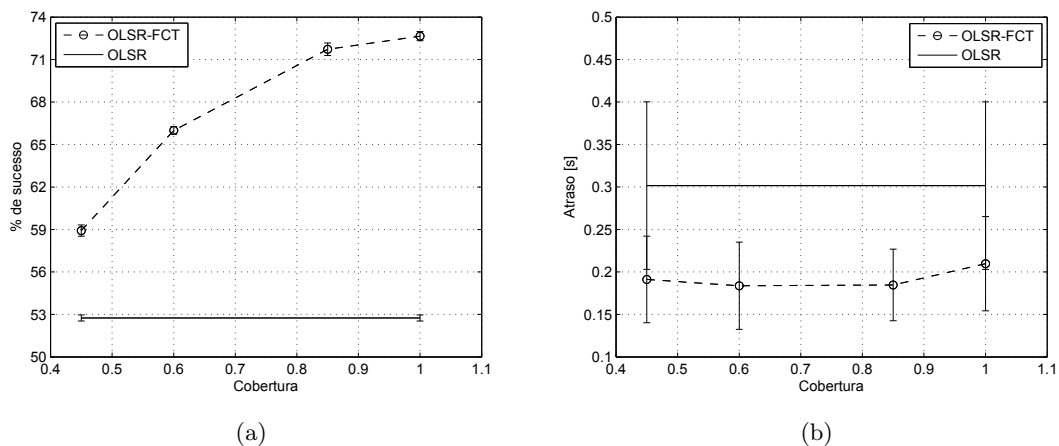


Figura 6.5: Percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), dos protocolo OLSR e OLSR-FCT, aplicado ao cenário Cen_{10v} , para diferentes valores de cobertura.

A Figura 6.6 apresenta o desempenho do protocolo OLSR-FCT nos quatro cenários de densidade de veículos. Como se pode observar, à medida que a densidade vai aumentando, a percentagem de sucesso de resolução dos pedidos de encaminhamento também aumenta. A diferença de percentagem de sucesso vai ficando mais acentuada quando se utiliza o protocolo OLSR-FCT com cobertura total. A Figura 6.6(a) ilustra que o protocolo OLSR-FCT deve ser utilizado apenas em cenários de elevada densidade, pois não é capaz de resolver situações de redes particionadas no mesmo sentido. Como tal, o método de eleição de nós MPR poderia ser modificado em cenários de baixa densidade, para permitir a eleição de veículos que viajam em sentido contrário como veículos MPR. Na Figura 6.6(b) é apresentado o atraso médio dos caminhos que, excluindo o cenário de densidade de 4 vizinhos (Cen_{4v}), descreve um aumento do atraso médio à medida que a densidade vai aumentando. Este facto deve-se ao aumento do tráfego *broadcast*, consequente do aumento do número de nós da rede, e aumento do tráfego de controlo, originado pelo aumento do número de veículos MPR.

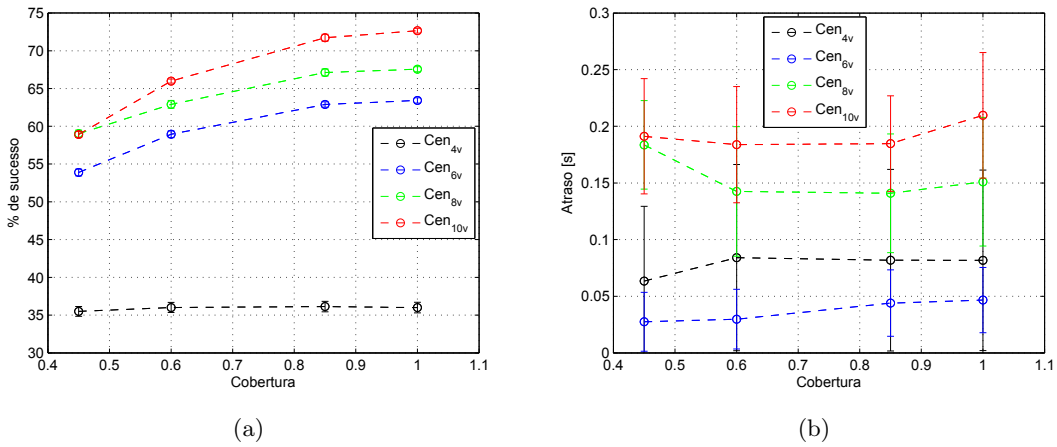


Figura 6.6: Comparação do desempenho do protocolo OLSR-FCT, relativamente à percentagem de sucesso de resolução dos pedidos de encaminhamento (a), e atraso médio dos caminhos (b), nos quatro cenários de simulação.

A Figura 6.7 apresenta o desempenho do protocolo OLSR-FCT no cenário de densidade de 6 vizinhos, face a várias parametrizações do limiar de estabilidade (k_{est}). Para um limiar de estabilidade inferior a 50 segundos, o método de eleição de veículos MPR do OLSR-FCT não consegue fazer a distinção entre os veículos que circulam no mesmo sentido e os veículos que circulam no sentido oposto, operando com um método de eleição de nós MPR bastante

semelhante ao do OLSR. Esta limitação reflecte-se nos resultados, pois a percentagem de sucesso dos pedidos de encaminhamento apresentado pelo OLSR-FCT, para este conjunto de valores, é inferior à obtida com um limiar de estabilidade parametrizado a 50 segundos. Pela Figura 6.7(a) é possível de observar que, por exemplo, o protocolo OLSR-FCT com $k_{est} = 35s$ exibe uma percentagem de sucesso de aproximadamente 47% (idêntica à obtida pelo protocolo OLSR (Figura 6.1)), face a 63% no caso de $k_{est} = 50s$, apresentando um acréscimo relativo de 31%. Quando o limiar de estabilidade é parametrizado acima dos 50 segundos, o desempenho do OLSR-FCT é também prejudicado, pois o método de eleição apenas selecciona como veículos MPR, aqueles que mantêm, uma ligação há mais de 65 segundos (no caso da Figura 6.7). Com esta parametrização é necessário que os veículos que viajem no mesmo sentido se mantenham em contacto durante bastante tempo, sendo ignoradas as ligações, que há partida, são teoricamente estáveis com apenas 50 segundos de duração.

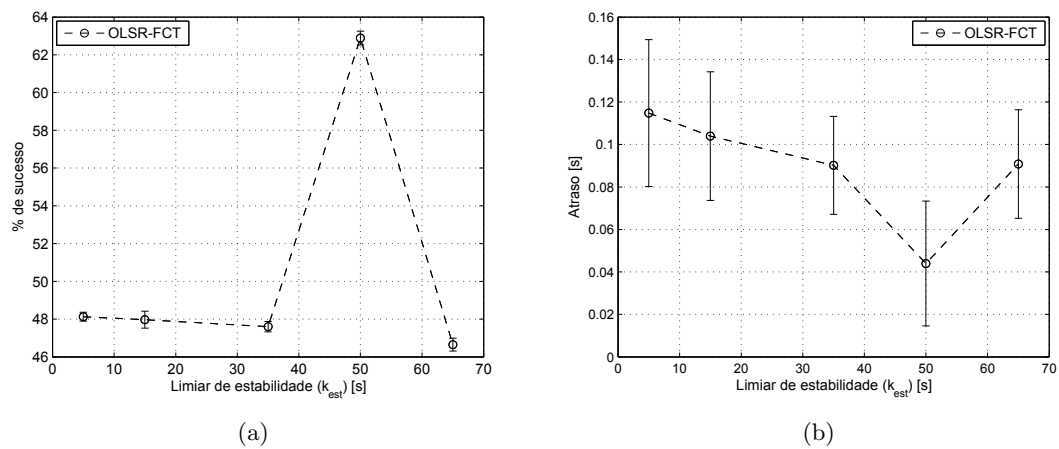


Figura 6.7: Influência do limiar da estabilidade (k_{est}) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 6 vizinhos (Cen_{6v}), com uma cobertura de 85%.

Relativamente ao atraso médio dos caminhos, a Figura 6.7(b) demonstra novamente que o valor mais vantajoso do limiar de estabilidade, por forma a melhorar o desempenho do protocolo OLSR-FCT, é de 50 segundos. Confrontando o pior valor de média de atraso obtido ($k_{est} = 5s$), com o atraso obtido com $k_{est} = 50s$ é de notar uma diminuição do tempo médio de atraso do caminho dos 115 ms para os 45 ms, o que representa um decréscimo relativo de aproximadamente 61%.

Por fim, é importante verificar o efeito da parametrização do T_O (*timeout* do *beacon*) no desempenho do protocolo OLSR-FCT. No final do Capítulo 4 chegou-se à conclusão que uma parametrização incorrecta deste parâmetro levaria a que o algoritmo de agrupamento de nós não obtivesse a visão correcta da rede, considerando que ligações fisicamente existentes deixassem de o ser, pelo facto do *beacon* chegar atrasado. Assim, foram testados os cenários de densidade de 6, 8 e 10 veículos com uma parametrização subdimensionada de $T_O = 2.5s$. O cenário de densidade de 4 vizinhos, por apresentar um fraco desempenho mesmo com o algoritmo de agrupamento de nós a obter uma visão correcta da rede, não foi testado com outro valor de T_O . As Figuras 6.8, 6.9 e 6.10 apresentam a influência do parâmetro de *timeout* do *beacon*, e em todos os cenários é possível de observar que o protocolo OLSR-FCT apresenta resultados superiores quando o algoritmo de agrupamento tem uma visão real da rede. Mais, em cenários de cobertura total, o protocolo OLSR-FCT com uma parametrização subdimensionada de T_O , vê o seu desempenho diminuir em comparação com o cenário de cobertura de 0.85. No que diz respeito ao atraso médio do caminho, a parametrização subdimensionada de T_O não apresenta grandes diferenças, existindo situações em que o atraso é menor (como por exemplo, no cenário Cen_{8v} com uma cobertura de 0.60), e outras em que o atraso sofre um ligeiro aumento (cenário Cen_{6v} com cobertura total).

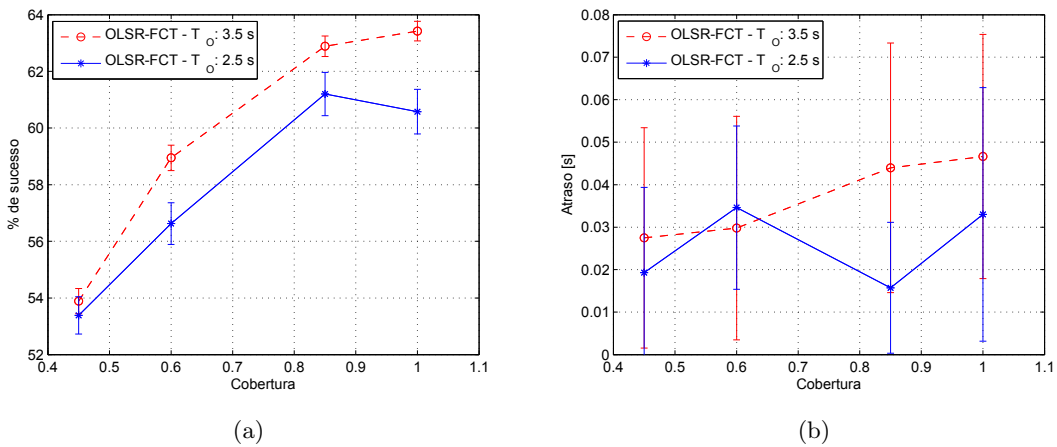


Figura 6.8: Influência do *timeout* do *beacon* (T_O) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 6 vizinhos (Cen_{6v}).

Analisado o desempenho do protocolo OLSR-FCT, face ao protocolo OLSR na presença

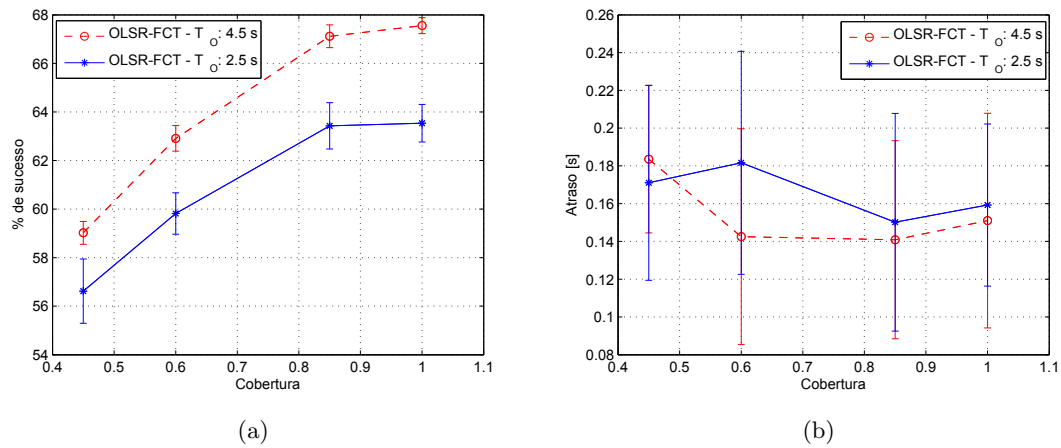


Figura 6.9: Influência do *timeout* do *beacon* (T_O) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 8 vizinhos (Cen_{8v}).

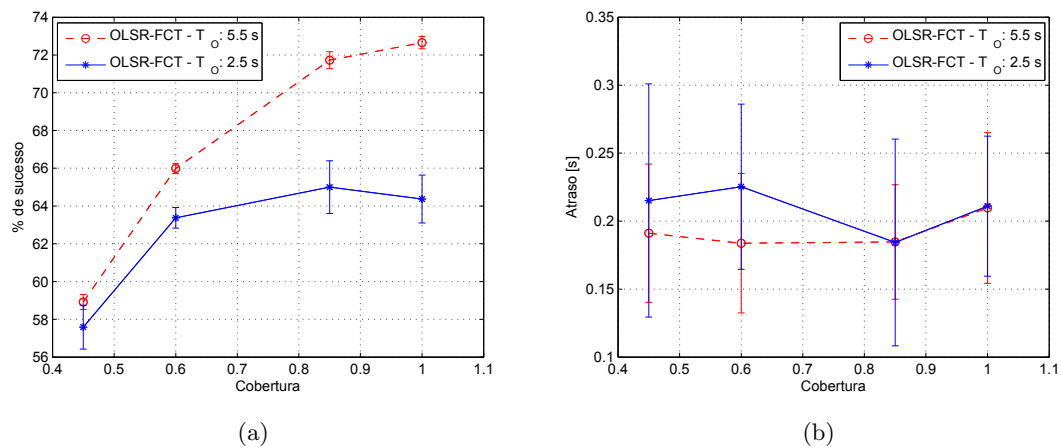


Figura 6.10: Influência do *timeout* do *beacon* (T_O) no desempenho do protocolo OLSR-FCT, no cenário de densidade média de 10 vizinhos (Cen_{10v}).

de tráfego em ambos os sentidos, é importante também confrontar o desempenho do protocolo OLSR-FCT ao do protocolo OLSR, quando existe trânsito a circular em apenas um sentido. Para tal, à Figura 1.1 apresentada na Secção 1.2, é adicionado o comportamento do protocolo OLSR-FCT no mesmo cenário de simulação, para o cenário com densidade média de 6 veículos (Cen_{6v}), obtendo-se a Figura 6.11. Em termos de percentagem de sucesso (Figura 6.11(a)), a aplicação do protocolo OLSR-FCT nos dois sentidos não consegue melhorar o desempenho do protocolo OLSR com apenas um sentido. Este facto pode ser justificado pelo aumento de tráfego *broadcast*, consequência do aumento do número de veículos. Analisando a Figura 6.11(b), que apresenta o atraso médio dos cami-

nhos, obtém-se que o desempenho do OLSR-FCT é superior ao do OLSR com apenas um sentido, sendo justificado pelo comportamento do algoritmo de identificação de ligações estáveis, que mesmo considerando tráfego a circular nos dois sentidos, só utiliza as ligações mais estáveis num sentido.

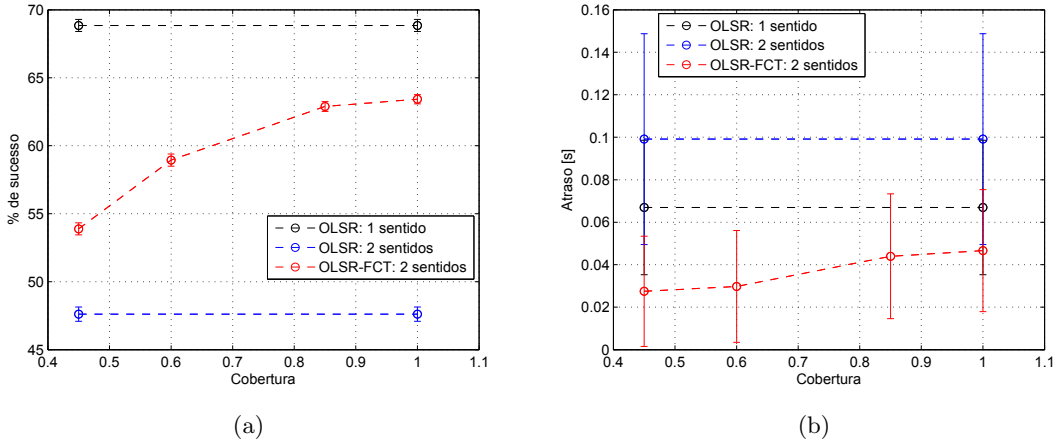


Figura 6.11: Comparação do desempenho do protocolo OLSR na presença de trânsito em um sentido, nos dois sentidos e do protocolo OLSR-FCT com trânsito nos dois sentidos.

Durante este capítulo analisou-se o desempenho do protocolo de encaminhamento OLSR-FCT. Em comparação com o protocolo OLSR, o protocolo OLSR-FCT apresentou melhores resultados em cenários de densidade média/elevada, tanto a nível de percentagem de sucesso de resolução de pedidos de encaminhamento, assim como no atraso dos caminhos. Estas melhorias são justificadas pela inclusão do algoritmo de identificação de ligações estáveis no processo de eleição dos nós MPR do protocolo OLSR. A utilização deste algoritmo veio permitir a identificação das ligações mais duradouras, modificando o método de eleição de nós MPR existente no OLSR. Com a eleição de nós MPR mais estáveis, reduz-se a probabilidade de se criarem caminhos que rapidamente ficarão desatualizados.

Observou-se ainda que, em cenários de densidade elevada, a quantidade de nós MPR eleitos poderá influenciar o atraso do caminho. As mensagens de controlo de topologia difundidas por este tipo de nó, em modo *broadcast*, originam um aumento no consumo da largura de banda, bastante limitada em redes sem fios. Como tal, a possível limitação da quantidade de nós MPR eleitos por cada nó, poderá ser uma mais valia em cenários que apresentem densidade bastante elevada.

Foi também analisada a influência do parâmetro que distingue as ligações estabelecidas entre veículos que viajam no mesmo sentido, das ligações entre veículos que viajam em sentidos opostos. Se este valor não estiver de acordo com a velocidade relativa dos veículos que viajam em sentidos opostos, a utilização do algoritmo de identificação de ligações não é vantajoso para o funcionamento do protocolo OLSR-FCT, que apresentou, neste caso, resultados idênticos aos do OLSR. Este facto vem reforçar ainda mais a ideia de que, a tarefa desempenhada pelo algoritmo de identificação das ligações mais estáveis, é a principal causa para o aumento do desempenho do protocolo OLSR-FCT num cenário de mobilidade de auto-estrada.

Capítulo 7

Conclusões

Neste último capítulo são realizadas algumas considerações finais sobre o trabalho realizado, onde são apresentadas as principais contribuições desta dissertação, bem como algumas direcções para trabalho futuro.

7.1 Considerações Finais

Nesta dissertação abordou-se um tema, que nos últimos anos tem sofrido um crescente interesse no seio da comunidade de investigação: a comunicação entre veículos. Sendo uma área de investigação que engloba vários aspectos, discutiu-se particularmente a possibilidade de criar um protocolo de encaminhamento capaz de apresentar um desempenho razoável, quando aplicado a um cenário de redes *ad-hoc* veiculares.

Observando-se os protocolos de encaminhamento existentes para redes *ad-hoc* móveis, pode-se concluir que estes não estão preparados para serem utilizados em redes veiculares, principalmente por não terem em consideração uma característica bastante importante das redes veiculares: a rápida e constante mudança topológica. Assim, e tendo em conta esta menos-valia, decidiu-se modificar o protocolo de encaminhamento OLSR, um protocolo com optimização de topologia e dos mais utilizados a nível de redes *ad-hoc*, adicionando um algoritmo de identificação de ligações consoante a sua duração (apresentado no Capítulo 4).

Em cenários de auto-estrada, os veículos que viajam no mesmo sentido tendem a man-

ter ligações mais prolongadas, relativamente às ligações estabelecidas entre veículos que viajam em diferentes sentidos. Foi a partir desta característica, que se decidiu adicionar ao método de eleição de nós MPR, característico do protocolo OLSR, um algoritmo de identificação de ligações. Graças a este algoritmo, é possível identificar quais as ligações existentes entre nós que viajam no mesmo sentido, e utilizá-las para realizar tanto encaminhamento como difusão de controlo de topologia, característico do OLSR.

Durante o Capítulo 6, foi possível observar que as modificações propostas ao protocolo OLSR, acabaram por oferecer uma melhoria no seu desempenho. A exclusiva utilização das ligações entre veículos que viajam no mesmo sentido de modo a realizar o encaminhamento, oferece rotas mais estáveis e, como tal, um encaminhamento mais coerente. Estes resultados foram demonstrados através de quatro cenários de simulação com diferentes densidades, e concluiu-se que à medida que a densidade de veículos aumenta, o protocolo OLSR-FCT demonstra um desempenho superior. O desempenho é aferido em termos de percentagem de sucesso de pedidos de encaminhamento, que chega a atingir um acréscimo relativo de 40%, no cenário de simulação de maior densidade, e na diminuição do tempo de atraso dos caminhos, que em alguns casos sofre um decréscimo de 70%.

Em jeito de conclusão, as modificações apresentadas nesta dissertação podem vir a trazer grandes valias para o desenvolvimento de um sistema de comunicação entre viaturas, ficando ainda a consciência de que muito trabalho há ainda a ser feito, como se sugere na secção seguinte.

7.2 Trabalho Futuro

As modificações ao protocolo OLSR discutidas nesta dissertação, apresentam algumas limitações, e como tal deveriam ser objecto de estudo no futuro. Como exemplo, de seguida são enumeradas algumas ideias que poderiam vir a enriquecer o trabalho realizado.

Para cenários de baixa densidade, o protocolo apresentado não oferece melhorias relativamente ao protocolo OLSR. Este facto deve-se à forma como o método de eleição de nós

MPR do protocolo OLSR-FCT está elaborado: apenas poderão ser eleitos como nós MPR aqueles que viajam no mesmo sentido, e apresentem uma relação de estabilidade acima do limiar. Ora, em cenários de baixa densidade, a probabilidade de não haver nós suficientes no mesmo sentido por forma a criar uma rede é maior. Assim, por forma a corrigir esta limitação, o método de eleição de nós MPR necessitaria de ser modificado, permitindo, em situações de baixa densidade, a eleição de nós MPR entre veículos que viajam em sentidos opostos.

O método de eleição de nós MPR do protocolo OLSR-FCT, depende em grande parte, do funcionamento do algoritmo de identificação de ligações. Se o algoritmo não conseguir distinguir as ligações estáveis da instáveis, o desempenho do protocolo OLSR-FCT é prejudicado. Um dos parâmetros mais importantes para o bom funcionamento do algoritmo de identificação é o *timeout* do *beacon* (T_O), pois define o período de tempo em que um *beacon* deverá ser recebido, de modo a que uma ligação seja considerada activa. No entanto, em cenários de densidade elevada, o tráfego *broadcast* gerado aumenta de tal forma, contribuindo para o aumento da probabilidade de existirem colisões entre pacotes e os *beacons*. Este acontecimento faz com que o algoritmo considere que as ligações físicas e estáveis não o sejam do ponto de vista lógico, obtendo uma visão errada da rede. Propõem-se então, que para cenários de alta densidade, e de acordo com o tráfego de *broadcast* gerado, o valor de *timeout* do *beacon* seja adaptado automaticamente, permitindo ao algoritmo de identificação de ligações manter uma visão real da rede, e assim aumentar o desempenho do protocolo OLSR.

Por último, e relativamente ao cenário de simulação, seria também interessante de verificar o desempenho dos protocolos, em cenários de redes híbridas (ou redes MESH), distribuindo pela auto-estrada uma quantidade razoável de estações base, que teriam como objectivo dar suporte às redes *ad-hoc* veiculares existentes.

Apêndice A

Aplicação Auxiliar para a Ferramenta SUMO

Neste apêndice é apresentada uma aplicação desenvolvida em linguagem MATLAB, durante a realização do projecto desta dissertação, que tem como função auxiliar o utilizador no desenvolvimento de cenários de mobilidade típica de auto-estrada, através da ferramenta SUMO. Esta aplicação apresenta grandes valias no desenvolvimento de cenários de auto-estrada, um processo que se torna bastante moroso quando se pretende obter um cenário de simulação com bastantes pontos de referência e com um número elevado de veículos. É também descrito o modo como se elabora um cenário de mobilidade, utilizando a ferramenta SUMO. Sugere-se que a leitura deste apêndice seja feita com a consulta do manual de utilizador da ferramenta SUMO [KR09].

Como descrito na Secção 3.2, a ferramenta SUMO é uma aplicação *open-source*, desenvolvida em linguagem JAVA, capaz de criar cenários de mobilidade, suportando micro-mobilidade. O método de criação de um cenário de mobilidade pode ser dividido em quatro etapas:

1. Informar a ferramenta SUMO de quantos pontos de referência, futuramente denominados por *nodes*, e de quantos caminhos que ligam os *nodes* (futuramente designado por *edges*), se pretendem.
2. Criação do mapa da rede através da informação dos *nodes* e dos *edges*.

3. Dar a indicação de quantos veículos se pretende introduzir na simulação, juntamente com a sua caracterização. É também necessário criar rotas de circulação para os veículos (*routes*), sendo assim possível associar cada veículo ou um conjunto de veículos, a uma *route* ou a um conjunto de *routes*. É ainda necessário associar a cada veículo um tempo de início de simulação.
4. Integrar o mapa da rede com a informação relativa aos veículos, obtendo-se finalmente o cenário de mobilidade.

A Figura A.1 ilustra de uma forma bastante simples os conceitos referidos no parágrafo anterior.

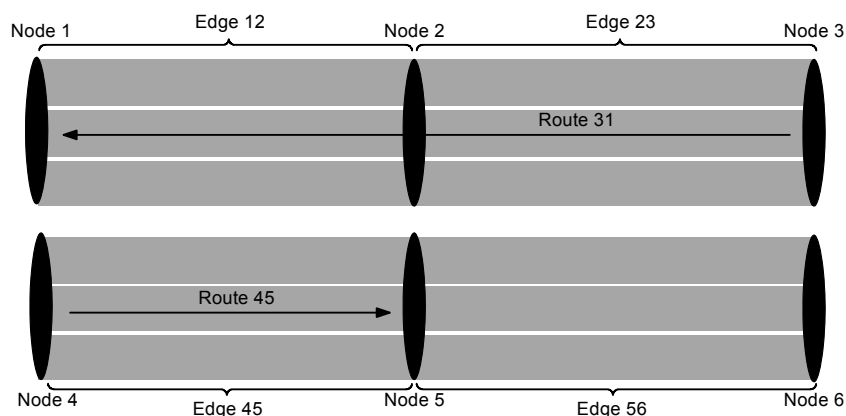


Figura A.1: Ilustração dos conceitos de *node*, *edge* e *route*, utilizados pela ferramenta SUMO.

Toda a informação que precisa de ser passada ao SUMO, descrita anteriormente, é feita através de ficheiros do tipo XML. Como tal, a ferramenta desenvolvida tem como função, criar de forma autónoma os três ficheiros XML necessários ao SUMO, indicando apenas: em forma de vector ordenado, a distribuição das posições dos *nodes* pretendida, onde posteriormente serão colocados os veículos no início da simulação; a posição do *node* inicial; a posição do *node* final; e um vector ordenado com os tempos de início de simulação de cada veículo. O tamanho do vector da distribuição das posições iniciais de veículos e o tamanho do vector com os tempos de início de simulação das posições, devolve a quantidade de veículos que serão iniciados no início da auto-estrada, e a quantidade de veículos que serão lançados com os tempos existentes no vector de tempos.

O primeiro passo no desenvolvimento de um cenário de mobilidade, através da ferramenta SUMO, passa pela criação dos *nodes*. A cada *node* é associado, entre outros parâmetros, um número de identificação *node id* e as suas posições *x* e *y*. O número de *nodes* necessários para desenhar um cenário depende das posições iniciais de cada veículo, e do percurso que os mesmo tenham de fazer, por exemplo: caso se pretenda obter um cenário de auto-estrada, em que no início da simulação estejam distribuídos espacialmente 20 veículos, será necessário criar 20 *nodes*. Na realidade, de modo a que o SUMO possa permitir a circulação de um veículo, são necessários no mínimo 2 *edges*, o que implica a criação de pelo menos 3 *nodes*. Logo, é sempre necessário acrescentar mais um *node*, que servirá de "estacionamento" para veículos que já terminaram o seu trajecto. O ficheiro XML, que contém a informação dos *nodes*, deverá ter o seguinte aspecto:

```
< nodes >
  < node id = "1" x = "0" y = "50" / >
  < node id = "2" x = "201" y = "50" / >
  < node id = "3" x = "480" y = "50" / >
  < node id = "4" x = "925" y = "50" / >
< /nodes >
```

A aplicação desenvolvida é capaz de elaborar o ficheiro XML que contém a informação dos *nodes*, sendo apenas necessário fornecer o vector da distribuição das posições. Esta primeira aplicação, utilizando o vector com a distribuição das posições, é capaz de criar um segmento de auto-estrada composto por todos os *nodes* que o compõem, e ainda pelos *nodes* inicial e final. Para além da criação do ficheiro, é ainda devolvido ao utilizador um vector constituído pelos *ids* dos *nodes* criados. Este vector será utilizado pela aplicação, na segunda etapa de funcionamento, que consiste em criar o ficheiro XML com a informação dos *edges*.

De modo a que a ferramenta SUMO possa criar o mapa da rede, para além do ficheiro XML dos *nodes*, necessita ainda do ficheiro que define e caracteriza os *edges*. A cada *edge* é associado um *id* de identificação do *edge*, um *node* de partida juntamente com um *node* de chegada, a quantidade de faixas que o compõem, a velocidade máxima de circulação

no *edge*, entre outros. Fornecendo à aplicação MATLAB o vector de identificação dos nós, resultado da primeira etapa de funcionamento desta aplicação, e sendo possível de parametrizar a quantidade de faixas desejadas assim como a velocidade máxima permitida, a aplicação procede à criação dos *edges* necessários para criar um segmento de auto-estrada, interligando os *nodes* sequencialmente pelos seus *ids*. Para além de criado o ficheiro, é devolvido ao utilizador um vector com a identificação dos *edges* criados, informação esta que será necessária na terceira e última etapa da aplicação. Como exemplo, para o ficheiro de *nodes* apresentado anteriormente, obtém-se o seguinte ficheiro de *edges*:

```
< edges >
< edge fromnode = "1" id = "1" tonode = "2" nolanes = "3" speed = "37" / >
< edge fromnode = "2" id = "2" tonode = "3" nolanes = "3" speed = "37" / >
< edge fromnode = "3" id = "3" tonode = "4" nolanes = "3" speed = "37" / >
< /edges >
```

A implementação destes ficheiros poderá parecer bastante simples, tornando esta aplicação fútil, no entanto, e como referido no início deste apêndice, na presença de uma grande quantidade *nodes* o uso desta aplicação poupa imenso tempo. Criados os dois ficheiros com a informação referente aos *nodes* e *edges*, é assim possível criar o mapa da rede. Esta tarefa é realizada com recurso a uma aplicação existente na ferramenta SUMO, que dá pelo nome de *netconvert*. A chamada a esta aplicação é feita da seguinte forma:

```
sumo-netconvert --xml-node-files = nodes.xml --xml-edge-files = edges.xml
--output - file = netowrk.net.xml
```

Por fim, a última etapa desta aplicação consiste na criação do ficheiro que contém toda a informação relativa aos *routes* e aos veículos. A ferramenta SUMO permite a criação de classes de veículos. Cada classe de veículo é caracterizada, entre outros, por um *id* de identificação, uma taxa de aceleração e desaceleração, comprimento dos veículos, velocidade máxima permitida para este tipo de veículos, e ainda um parâmetro que expressa a imperfeição do condutor. As *routes*, necessárias para indicar o percurso que um veículo irá percorrer, são definidas também por um *id* de identificação, e ainda por um conjunto de *edges* que definem a rota. Criadas as classes de veículos e as rotas necessárias para os

veículos circularem, podem finalmente ser criados os veículos. A cada veículo é associado um *id* de identificação, uma classe de veículo, uma *route* e o tempo de início de simulação. De seguida é apresentado um exemplo de um ficheiro XML que contém a informação descrita anteriormente.

```
< routes >
  < vtype id = "Type1" accel = "3.50" decel = "3.50" length = "4.00" maxspeed = "27.80" sigma = "0.00" / >
  < vtype id = "Type2" accel = "2.50" decel = "3.00" length = "5.00" maxspeed = "26.00" sigma = "0.00" / >
  < vtype id = "Type3" accel = "1.50" decel = "2.00" length = "8.00" maxspeed = "20.00" sigma = "0.00" / >

  < route id = "route1" multi.ref = "x" edges = "1 2 3" / >
  < route id = "route2" multi.ref = "x" edges = "2 3" / >

  < vehicle id = "1" type = "Type2" route = "route1" depart = "0" / >
  < vehicle id = "2" type = "Type1" route = "route2" depart = "0" / >
  < vehicle id = "3" type = "Type3" route = "route1" depart = "20" / >
< /routes >
```

De modo a que a aplicação desenvolvida crie o ficheiro XML, é necessário fornecer o vector de tempos de partida dos veículos, assim como o vector de identificação dos *edges*, resultado da segunda etapa desta aplicação. Podem ainda ser parametrizadas as classes de veículos desejadas. A aplicação, para além de criar as classes de veículos, vai criar tantas *routes* de acordo com o tamanho do vector de *edges*, ou seja, para cada veículo iniciado no meio da auto-estrada é criada uma *route* até ao final da mesma. É criada ainda uma *route* adicional composta por todos os *edges* do mesmo sentido, o que origina uma *route* que vai desde o início ao final da auto-estrada. Os restantes veículos, os que irão percorrer a auto-estrada completa, serão criados consoante o tamanho do vector de tempos de partida. Assim, pode-se afirmar que o número total de veículos da simulação é dado pela soma do número de *edges* mais o tamanho do vector de tempos de partida.

Esta aplicação tem ainda a vantagem de elaborar os ficheiros necessários para a criação de *nodes*, *edges*, *routes* e veículos que irão percorrer a auto-estrada em sentido contrário, sem criar conflitos de identificação. Ou seja, esta aplicação identifica os veículos de um sentido de forma sequencial, identificando depois os veículos que circulam em sentido contrário. Finalmente, é possível criar o cenário de mobilidade, através da ferramenta principal do SUMO que deve ser chamada da seguinte forma:

```
sumo --net - file = network.net.xml --route - files = routes.xml --begin INT --end INT
```

Os parâmetros *begin* e *end* referem-se ao tempo de início e de fim da simulação respectivamente. O cenário de mobilidade gerado é considerado pelos autores da ferramenta SUMO como sendo uma versão *dump*, pois ainda não está devidamente configurada para ser utilizada em nenhum simulador de redes. Para tal, é necessária a utilização de outra aplicação da ferramenta SUMO denominada de *traceExporter*. Com a utilização desta aplicação é possível transformar a versão *dump* do cenário de mobilidade, numa versão que possa ser utilizada pelo simulador *ns-2*. Para além disso, esta aplicação fornece ainda a informação necessária para a configuração do *script* de simulação do *ns-2*, assim como um ficheiro capaz de identificar quando é que os nós estão activos/inactivos. Este ficheiro é bastante importante, de modo a inutilizar os agentes dos veículos quando estes ainda não iniciaram o seu andamento, ou quando já chegaram ao final da auto-estrada.

No entanto, a utilização da aplicação *traceExporter* tem um senão: sempre que a versão final do cenário de mobilidade é criada, a identificação dos veículos parametrizada no ficheiro XML é alterada, tornando assim a identificação dos veículos não sequencial, o que torna mais complicado de diferenciar quais os veículos que circulam no mesmo sentido, dos que circulam em sentido contrário. Como tal, procedeu-se à modificação do ficheiro *MobilityWriter*, pertencente ao *package* da aplicação *traceExporter*, alterando a forma como é escrita a versão final do cenário de mobilidade, mais propriamente a identificação dos veículos.

Nas páginas seguintes é apresentada a aplicação desenvolvida. Esta encontra-se dividida em 3 ficheiros, dependendo dos ficheiros XML que se desejem criar.

```

% Function that will generate the nodes.xml and return an array with all id nodes
function[final_nodes, final_nodes_inv] = create_nodes_all(array_nodes, tbegin, tend)
y = 50;
y_inv = 80;

fid = fopen('nodes.xml', 'wt');
fprintf(fid, '< nodes > \n');

fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', 1, tbegin, y); final_nodes(1) = 1;

for i = 1 : length(array_nodes)
    fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', i + 1, array_nodes(i), y); final_nodes(i + 1) = i + 1;
end
fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', i + 2, tend, y); final_nodes(i + 2) = i + 2;
fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', i + 3, tend + 10, y); final_nodes(i + 3) = i + 3;

fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', 4 + length(array_nodes), tbegin - 10, y_inv);
final_nodes_inv(1) = 4 + length(array_nodes);
fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', 5 + length(array_nodes), tbegin, y_inv);
final_nodes_inv(2) = 5 + length(array_nodes);
for i = 1 : length(array_nodes)
    fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', i + 5 + length(array_nodes), array_nodes(i), y_inv);
    final_nodes_inv(i + 2) = i + 5 + length(array_nodes);
end
fprintf(fid, '< nodeid = "%i" x = "%i" y = "%i" / > \n', i + 6 + length(array_nodes), tend, y_inv);
final_nodes_inv(i + 3) = i + 6 + length(array_nodes);

fprintf(fid, '< nodeid = "fbegin" x = "%i" y = "%i" / > \n', tbegin, 10);
fprintf(fid, '< nodeid = "fend" x = "%i" y = "%i" / > \n', tend + 10, 10);

fprintf(fid, '< /nodes >');
fclose(fid);

```

```

% Function that will generate the edges.xml and return an array with all id edges
function[edge, edge_inv] = create_edges_all(final_nodes, final_nodes_inv)
nolanes = 3;
speed = 37.0;

fid = fopen('edges.xml', 'wt');
fprintf(fid, '< edges > \n');

for i = 1 : length(final_nodes) - 1
    fprintf(fid, '< edgefromnode = "%i" id = "%i" tonode = "%i" nolanes = "%i" speed = "%d" / > \n',
        final_nodes(i), i, final_nodes(i + 1), nolanes, speed);
    edge(i) = i;
end

i = length(final_nodes) - 1;
aux = length(final_nodes);
while(i > 0)
    fprintf(fid, '< edgefromnode = "%i" id = "%i" tonode = "%i" nolanes = "%i" speed = "%d" / > \n',
        final_nodes_inv(i + 1), aux, final_nodes_inv(i), nolanes, speed);
    edge_inv(i) = aux;
    aux = aux + 1;
    i = i - 1;
end
edge_inv = sort(edge_inv);

fprintf(fid, '< edgefromnode = "fbegin" id = "fake" tonode = "fend" nolanes = "%i" speed = "%d" / > \n',
    nolanes, speed);

fprintf(fid, '< /edges >');
fclose(fid);

```

```

% Function that will generate the routes.rou.xml using the input array of the edges id
function create_routes_all(edge, edge_inv, array_depart)

vtype(1).id = 'Type1';
vtype(1).accel = 3.5;
vtype(1).decel = 3.5;
vtype(1).length = 4.0;
vtype(1).maxspeed = 27.8;
vtype(1).sigma = 0.0;
vtype(2).id = 'Type2';
vtype(2).accel = 2.5;
vtype(2).decel = 3.0;
vtype(2).length = 5.0;
vtype(2).maxspeed = 26.0;
vtype(2).sigma = 0.0;
vtype(3).id = 'Type3';
vtype(3).accel = 1.5;
vtype(3).decel = 2.0;
vtype(3).length = 8.0;
vtype(3).maxspeed = 20.0;
vtype(3).sigma = 0.0;
aux_tot_vehicle = 0;

fid = fopen('routes.rou.xml', 'wt');
fprintf(fid, '< routes > \n');

for i = 1 : length(vtype)
    fprintf(fid, '< vtype id = "%s" accel = "%2.2f" decel = "%2.2f" length = "%2.2f" '
        maxspeed = "%2.2f" sigma = "%2.2f" / > \n', vtype(i).id, vtype(i).accel, vtype(i).decel,
        vtype(i).length, vtype(i).maxspeed, vtype(i).sigma);
end

for i = 1 : length(edge) - 1
    fprintf(fid, '< route id = "route%i" multi-ref = "x" edges = "', i);
    for j = i : length(edge)
        fprintf(fid, '%i', j);
    end
    fprintf(fid, '" / > \n');
    route(i) = i;
end

for l = 1 : length(edge_inv) - 1
    fprintf(fid, '< route id = "route%i" multi-ref = "x" edges = "', l + length(route));
    for j = l : length(edge_inv)
        fprintf(fid, '%i', edge_inv(j));
    end
    fprintf(fid, '" / > \n');
    route_inv(l) = l + length(route);
end

for i = 1 : length(route) - 1
    aux = rand;
    if aux <= 0.6
        type = 1;
    elseif aux > 0.6 && aux <= 0.85
        type = 2;
    else
        type = 3;
    end
    fprintf(fid, '< vehicle id = "%i" type = "Type%i" route = "route%i" depart = "0" / > \n', i, type, route(i));
    aux_tot_vehicle = aux_tot_vehicle + 1;
end

for i = 1 : length(array_depart)
    aux = rand;
    if aux <= 0.6
        type = 1;
    elseif aux > 0.6 && aux <= 0.85
        type = 2;
    else
        type = 3;
    end
    aux_tot_vehicle = aux_tot_vehicle + 1;
    fprintf(fid, '< vehicle id = "%i" type = "Type%i" route = "route1" depart = "%i" / > \n',
        aux_tot_vehicle, type, array_depart(i));
end

for i = 1 : length(route_inv) - 1
    aux = rand;
    if aux <= 0.6
        type = 1;
    elseif aux > 0.6 && aux <= 0.85
        type = 2;
    else
        type = 3;
    end
    aux_tot_vehicle = aux_tot_vehicle + 1;
    fprintf(fid, '< vehicle id = "%i" type = "Type%i" route = "route%i" depart = "0" / > \n',
        aux_tot_vehicle, type, route_inv(i));
end

for i = 1 : length(array_depart)
    aux = rand;
    if aux <= 0.6
        type = 1;
    elseif aux > 0.6 && aux <= 0.85
        type = 2;
    else
        type = 3;
    end
    aux_tot_vehicle = aux_tot_vehicle + 1;
    fprintf(fid, '< vehicle id = "%i" type = "Type%i" route = "route%i" depart = "%i" / > \n',
        aux_tot_vehicle, type, route_inv(1), array_depart(i));
end

fprintf(fid, '< / routes >');
fclose(fid);

```

Apêndice B

Script de Simulação do *ns-2*

Neste apêndice é apresentada a estrutura dos *scripts* do simulador de redes *ns-2*, utilizados para avaliar o desempenho dos protocolos OLSR e OLSR modificado, segundo as propostas apresentadas no Capítulo 5. Como os cenários de mobilidade gerados são bastante característicos, na medida em que alguns veículos são colocados no início de cada sentido da auto-estrada, recebendo depois ordem para circular, houve a necessidade de criar ficheiros auxiliares para activar/desactivar o protocolo de encaminhamento, de modo a não influenciar as simulações.

No início do *script* de simulação são definidos alguns parâmetros, relacionados com o nível físico, o sub-nível MAC e o nível de rede. Começa-se por parametrizar o tipo de canal, assim como o modelo de propagação rádio e o tipo de antena, já que se trata de um cenário de redes sem fios (linhas 1 a 3). De seguida é seleccionado o tipo de fila de espera a utilizar, assim como o número máximo de pacotes suportados pela fila de espera (linha 5 e 6). Por fim, é parametrizada a interface de rede, e seleccionada a norma IEEE 802.11 para operar a nível MAC e o protocolo OLSR como protocolo de encaminhamento (linhas 7 a 9).

Na linha 10 é carregado o ficheiro que contém a informação sobre a mobilidade dos nós. Nas linhas 11 e 12 são carregados os ficheiros responsáveis pela activação/desactivação do protocolo de encaminhamento, assim como da aplicação geradora de tráfego, ambos associados a cada nó, obtidos através da aplicação *TraceExporter* da ferramenta SUMO. Entre

as linhas 13 e 18 é definido o número de nós da simulação, assim como os limites do cenário e o tempo de duração da simulação. As linhas 16 e 17 apresentam um *offset* aos limites do cenário de simulação, indicados pela ferramenta que gerou o cenário de mobilidade, a ferramenta SUMO. A linha 19 afecta o valor da semente do gerador de números aleatórios, necessária na aplicação geradora de tráfego, e que é alterada cada vez que uma simulação é iniciada. Como a aplicação geradora de tráfego escolhe aleatoriamente o nó de destino, ao ser modificado o valor da semente, é afectada a aleatoriedade da escolha dos nós de destino, de simulação para simulação. Na linha 20 é definido o número de veículos que circulam apenas num sentido, permitindo à aplicação geradora de tráfego escolher como nó de destino os veículos que apenas circulam no mesmo sentido.

set opt(chan)	Channel/WirelessChannel	<i>Linha 1</i>
set opt(prop)	Propagation/TwoRayGround	<i>Linha 2</i>
set opt(ant)	Antenna/OmniAntenna	<i>Linha 3</i>
set opt(ll)	LL	<i>Linha 4</i>
set opt(ifq)	Queue/DropTail/PriQueue	<i>Linha 5</i>
set opt(ifqlen)	50	<i>Linha 6</i>
set opt(netif)	Phy/WirelessPhy	<i>Linha 7</i>
set opt(mac)	Mac/802.11	<i>Linha 8</i>
set opt(rp)	OLSR	<i>Linha 9</i>
set opt(aa)	"activity_app"	<i>Linha 11</i>
set opt(an)	"activity_nodes"	<i>Linha 12</i>
set opt(nn)	80.0	<i>Linha 13</i>
set opt(x)	10020	<i>Linha 14</i>
set opt(y)	79	<i>Linha 15</i>
set opt(min-x)	0	<i>Linha 16</i>
set opt(min-y)	-9	<i>Linha 17</i>
set opt(stop)	1000.0	<i>Linha 18</i>
set opt(semente)	289	<i>Linha 19</i>
set opt(nodes.1way)	40	<i>Linha 20</i>

As linhas 21 e 22 são utilizadas para definir o alcance rádio de cada veículo como 1000 *m*. A linha 23 parametriza a taxa de transmissão de tráfego de *broadcast* a 11 *Mbps*, enquanto que na linha 22 é definido uma taxa de transmissão de tráfego *unicast* a 2 *Mbps*. Na linha 25, é parametrizado o tamanho mínimo de cada pacote enviado, para que seja necessário o envio de mensagens RTS/CTS: neste caso como o tamanho dos pacotes de

tráfego não excede os 3000 *bytes*, não existe troca de mensagens RTS/CTS. Referente ao protocolo de encaminhamento OLSR, é parametrizado o período de envio de mensagens do tipo HELLO com 1 *s* (linha 26) e mensagens de controlo, do tipo TC, a 2 *s* (linha 27).

Phy/WirelessPhy set RXThresh_	1.42681e-12	<i>Linha 21</i>
Phy/WirelessPhy set CStresh_	1.559e-13	<i>Linha 22</i>
Mac/802.11 set dataRate_	11.0e6	<i>Linha 23</i>
Mac/802.11 set basicRate_	2.0e6	<i>Linha 24</i>
Mac/802.11 set RTSThreshold_	3000	<i>Linha 25</i>
Agent/OLSR set hello.ival_	1;	<i>Linha 26</i>
Agent/OLSR set tc.ival_	2;	<i>Linha 27</i>

De seguida é instanciado o simulador *ns-2* (linha 28) e definido um ficheiro de *output* para obter a informação resultante das simulações (linhas 29 e 30). Nas linhas 31 e 32 é criada a topologia da rede, de acordo com os limites do cenário parametrizados anteriormente.

set ns_	[new Simulator]	<i>Linha 28</i>
set tracefd	[open traces/OLSR/output0.tr w]	<i>Linha 29</i>
\$ns_ trace-all \$tracefd		<i>Linha 30</i>
set topo	[new Topography]	<i>Linha 31</i>
\$topo load_flatgrid \$opt(x) \$opt(y)		<i>Linha 32</i>

Entre as linhas 33 e 48 é realizada a configuração de cada nó, associando a cada um as parametrizações definidas no início do *script*, enquanto que na linha 49 são instanciados os nós na simulação.

Na linha 50 é carregado o ficheiro de mobilidade, enquanto que na linha 51 é carregado o ficheiro responsável pela activação/desactivação do protocolo de encaminhamento associado ao nó. A linha 52 é responsável por instanciar a aplicação de tráfego de ficheiros entre os nós, com a indicação da quantidade de nós que compõem a simulação. Na linha 53, é carregado na aplicação de tráfego, o ficheiro que tem o agendamento do tráfego associado a cada nó. Entre as linhas 54 e 62 existe toda a informação relativa à associação entre cada

nó ao seu agente de tráfego, que por sua vez, está associado a uma aplicação de tráfego.

set chan_1_ [new \$opt(chan)]	<i>Linha 33</i>
\$ns_ node-config-adhocRouting \$opt(rp)	<i>Linha 34</i>
-llType \$opt(ll)	<i>Linha 35</i>
-macType \$opt(mac)	<i>Linha 36</i>
-ifqType \$opt(ifq)	<i>Linha 37</i>
-ifqLen \$opt(ifqlen)	<i>Linha 38</i>
-antType \$opt(ant)	<i>Linha 39</i>
-propType \$opt(prop)	<i>Linha 40</i>
-phyType \$opt(netif)	<i>Linha 41</i>
-topoInstance \$topo	<i>Linha 42</i>
-agentTrace ON	<i>Linha 43</i>
-routerTrace OFF	<i>Linha 44</i>
-macTrace OFF	<i>Linha 45</i>
-wiredRouting OFF	<i>Linha 46</i>
-movementTrace OFF	<i>Linha 47</i>
-channel \$chan_1_	<i>Linha 48</i>
for {set i 0} {\$i < \$opt(nn)} {incr i} { set node_(\$i) [\$ns_ node]; }	<i>Linha 49</i>

O agente de tráfego é responsável pelos pedidos de encaminhamento, enquanto que a aplicação de tráfego tem como função agendar, e de seguida, escolher aleatoriamente um nó de destino para o pedido de encaminhamento, no tempo que está agendado no ficheiro carregado na linha 53. Nas linhas 61 e 62 respectivamente, são associados à aplicação de tráfego, a variável responsável pela aleatoriedade da escolha do nó de destino e o número de nós que viajam num sentido, que limitam a escolha dos nós de destino no pedido de encaminhamento. Como a aplicação de tráfego já foi associada a cada nó, é possível então carregar o ficheiro responsável pela activação/desactivação desta aplicação (linha 63).

Na linha 64 é dada a informação a cada nó de quando termina a simulação, enquanto que na linha 65 é informado ao simulador *ns-2* quando deverá invocar o procedimento *stop* descrito na linha 66. Este procedimento é responsável por escrever no ficheiro de *output* a informação que estiver em *buffer* quando a simulação termina. Por fim, a linha 66 é

responsável por dar início à simulação.

source	\$opt(sc)	<i>Linha 50</i>
source	\$opt(an)	<i>Linha 51</i>
set	bSys [new TrafficController 80]	<i>Linha 52</i>
\$bSys	charge-traffic traffic_n40_avg2_offset.txt	<i>Linha 53</i>
for {set i 0} {	\$i < \$opt(nn)} {incr i 1} {	<i>Linha 54</i>
	set gen(\$i) [new Agent/TrafficAgent]	<i>Linha 55</i>
	\$ns_ attach-agent \$node_(\$i) \$gen(\$i)	<i>Linha 56</i>
	\$node_(\$i) attach \$gen(\$i) 333	<i>Linha 57</i>
	set app_(\$i) [new Application/P2PApp];	<i>Linha 58</i>
	\$app_(\$i) attach-traffic-controller \$bSys	<i>Linha 59</i>
	\$app_(\$i) attach-agent \$gen(\$i)	<i>Linha 60</i>
	\$app_(\$i) seed \$opt(semente)	<i>Linha 61</i>
	\$app_(\$i) nodes_1way \$opt(nodes_1way)	<i>Linha 62</i>
}		
source	\$opt(aa)	<i>Linha 63</i>

for {set i 0} {	\$i < \$opt(nn)} {incr i} {	<i>Linha 64</i>
	\$ns_ at \$opt(stop) "\$node_(\$i) reset"	
}		
\$ns_ at \$opt(stop) "stop"		<i>Linha 65</i>
proc stop {} {		<i>Linha 66</i>
	global ns_ tracefd	
	\$ns_ flush-trace	
	close \$tracefd	
}		
\$ns_ run		<i>Linha 66</i>

Apêndice C

Publicações

Neste apêndice são apresentados os artigos realizados ao longo desta dissertação. O primeiro, intitulado "Controlo de Topologia de Redes *ad hoc* Veiculares em cenários de Auto-estradas", foi submetido na Conferência de Redes de Computadores no ano de 2009 (CRC2009), enquanto que o segundo, intitulado "Improving Routing Performance in High Mobility and High Density *ad hoc* Vehicular Networks", encontra-se em fase de revisão numa conferência internacional (WCNC 2010). Ambos os artigos têm por base todo o trabalho realizado ao longo do projecto desta dissertação.

Controlo de Topologia de Redes *ad hoc* Veiculares em cenários de Auto-estradas

M. Luís , R. Oliveira , L. Bernardo , P. Pinto
FCT-UNL, Universidade Nova de Lisboa, Portugal
UNINOVA, Monte de Caparica, Portugal

Sumário—Este trabalho descreve um método de controlo de topologia em redes móveis formadas por veículos que circulam num cenário de auto-estrada. O controlo de topologia pretende realizar a caracterização dos veículos (nós) que formam a rede *ad hoc*, de modo a identificar os nós que circulam em cada um dos sentidos. Dado que a mobilidade dos nós em cada um dos sentidos da auto-estrada tende a cumprir o modelo de mobilidade microscópico denominado “*follow-the-leader*”, as viaturas de uma dada classe de tráfego que circulam num dado sentido apresentam uma velocidade relativa muito semelhante à dos seus nós vizinhos. Este facto é explorado por um algoritmo de controlo de topologia, o qual agrupa os nós em grupos de difusão de informação baseando-se indirectamente na sua velocidade relativa. A utilidade destes grupos é demonstrada através da integração do algoritmo no protocolo de encaminhamento OLSR. Comparam-se as estatísticas da taxa de sucesso de encaminhamento e do tempo necessário a visitar o nó de destino (atraso do caminho), caso exista caminho para esse nó. Verifica-se que a utilização do algoritmo de controlo de topologia e a realização de pequenas alterações no algoritmo de difusão da topologia, apresenta melhores resultados quer na taxa de sucesso de encaminhamento quer no atraso do caminho.

Palavras-chave: Controlo de Topologia, Protocolos de Encaminhamento, Redes *ad hoc* Veiculares.

I. INTRODUÇÃO

Este artigo propõe uma optimização para os protocolos de encaminhamento baseados em estado de linha, em redes *ad hoc* veiculares (VANET - *Vehicular Ad-hoc NETwork*), móveis e potencialmente instáveis (a instabilidade refere-se às situações onde a conectividade entre dois nós é interrompida devido à mobilidade dos nós). Estes protocolos realizam a descoberta de vizinhos através da difusão periódica de um pacote HELLO na rede sem fios, e difundem na rede a informação topológica de forma periódica ou em resposta a mudanças topológicas. Desta forma, cada nó obtém uma cópia local da topologia e calcula localmente a tabela de encaminhamento.

O cenário de uma auto-estrada define uma rede com características que são particularmente destrutivas para os protocolos de encaminhamento existentes [1] [2]. O conjunto de carros em cada faixa exibe um comportamento relativamente estável, nomeadamente se os veículos pertencerem à mesma classe de tráfego (com os mesmos parâmetros de mobilidade). No entanto, a velocidade relativa entre dois veículos que circulem em diferentes sentidos pode facilmente atingir os 240Km/h, originando tempos de duração de conectividade entre eles de aproximadamente 30 segundos (considerando um alcance de

rádio de 1 Km). Este valor é proporcional ao alcance de rádio considerado. Uma das consequências do baixo valor do tempo de duração da conectividade é a necessidade de ter um período de difusão de HELLO igualmente baixo (da ordem de grandeza do segundo), e a detecção frequente de mudanças de topologia. As mudanças de topologia originam a disseminação na rede de actualizações de topologia. Em [3] mostrou-se que a elevada utilização de tráfego de difusão numa rede 802.11 pode levar à sua destruição, originando a chamada tempestade de *broadcast* [4]. Desta forma, é extremamente importante limitar o tráfego de sinalização do protocolo de encaminhamento de forma a deixar que algum tráfego de aplicações possa usar a rede.

O protocolo OLSR (*Optimized Link-State Routing*) [2] propõe a estratégia de limitar a tarefa da difusão da informação topológica a um conjunto limitado de veículos (os nós retransmissores multiponto - MPR *Multipoint Relay*). No entanto, a selecção dos nós MPR não utiliza nenhum critério relacionado com a velocidade relativa entre nós. Usa como critério principal o número de vizinhos de cada nó, permitindo também aos nós anunciarem a sua prontidão (*willingness*) para serem MPR. Desta forma, embora reduza o número de nós a enviar actualizações topológicas, não reduz o ritmo a que estas são geradas.

Neste artigo é proposto uma nova abordagem para realizar a selecção dos nós MPR, que tem em conta a estabilidade relativa entre os vários nós. É proposto um algoritmo de controlo de topologia que mede a estabilidade relativa entre os vários nós, e que fornece os parâmetros necessários para se realizar uma escolha de MPR que exclui os nós com maiores velocidades relativas, reduzindo significativamente o número de actualizações. Por outro lado, sacrifica a total cobertura da rede em situações de carga de sinalização muito elevada, excluindo nós com menor estabilidade relativa do conjunto de MPRs. Desta forma, mesmo em situações críticas, garante-se que existe sempre algum tráfego útil que pode ser usado pelas aplicações. Na secção seguinte é apresentado o conceito de estabilidade usado e o algoritmo proposto para realizar o controlo de topologia. Na secção III são apresentadas as modificações realizadas ao protocolo OLSR, nomeadamente o algoritmo de selecção de MPR. Depois, na secção IV, são apresentados resultados de desempenho medidos com recurso a simulações onde se mostra a melhoria de desempenho. Finalmente, na secção V são apresentadas as conclusões e as propostas para trabalho futuro.

II. CONTROLO DE TOPOLOGIA

O controlo de topologia proposto baseia-se no agrupamento de nós baseado em estabilidade de ligações. É proposto um algoritmo de baixa complexidade para definir um conjunto de nós (os líderes de grupo de *broadcast*) que, pelas suas características de mobilidade, podem ser utilizados como nós de difusão das actualizações de topologia, minimizando o número de nós que realizam a disseminação. A rede é dividida em vários grupos, e o algoritmo agrupa nós estáveis em grupos de nós que se encontram dentro do alcance rádio.

A. Agrupamento de nós baseado em estabilidade das ligações

Cada nó elege um LGB (líder de grupo de *broadcast*), utilizando o algoritmo de agrupamento. Para apresentar o algoritmo, começa-se por definir alguns conceitos base. São assumidas conexões bidireccionais entre dois nós.

A vizinhança é definida, tendo em conta que existe um meio lógico de identificação dos vizinhos físicos (identificação através do envio/recepção de *beacons*). Seja N o conjunto de nós de quem o nó n recebe *beacons* durante um determinado intervalo de tempo. N representa o conjunto de nós vizinhos lógicos de n . A noção de vizinhança lógica é mais restrita que a vizinhança física, pois dois nós podem estar dentro do alcance rádio, mas, devido à não recepção de *beacons* num dado intervalo de tempo, a relação de vizinhança lógica pode não se verificar.

Todo o nó n que pertence ao conjunto N dos nós da rede, pode eleger um nó vizinho denominado líder do grupo de *broadcast* (BGL). O nó LGB eleito pelo nó n é representado por n_{LGB} . Dado o conjunto de nós N , a condição $n_{LGB} \in N$ impõe que todos os nós pertencentes a N escolham o mesmo nó LGB (nó n_{LGB}). Nestas condições um grupo de *broadcast* (GB) é definido através do conjunto N .

Cada nó envia periodicamente *beacons*, os quais utilizam tramas do tipo *broadcast* transmitidas com frequência f . Cada *beacon* contém a identificação do nó que o envia, a identificação do LGB eleito e um identificador único do *beacon*. Considera-se o instante t em que um nó n recebe o primeiro *beacon* transmitido pelo seu vizinho n_i , estabelecendo-se uma ligação lógica entre os nós n e n_i .

Define-se, de seguida, o conceito de estabilidade da ligação lógica. Um nó n que receba *beacons* de um nó vizinho n_i possui um determinado valor de estabilidade S_{n,n_i} com esse vizinho. A estabilidade S_{n,n_i} afere a duração da relação de vizinhança entre os nós. O valor de S_{n,n_i} , determinado pelo nó n no instante temporal t , é dado pela expressão $S_{n,n_i} = \frac{t}{div}$, onde div representa a operação de divisão inteira entre t e div .

Os nós mantêm uma tabela de *beacons* (também denominada tabela de vizinhos lógicos) que descreve as ligações lógicas desse nó com os seus nós vizinhos. Os nós vizinhos são representados por cada um dos registos da tabela de *beacons*, o qual inclui: o endereço do nó vizinho que envia o *beacon* (n_i); um campo temporário utilizado pelo algoritmo de eleição do LGB contendo o valor da estabilidade da ligação

com esse vizinho (n_i) sendo actualizado no instante em que é executado o algoritmo; o líder de grupo (n_{LGB}) eleito por esse vizinho; o instante t em que foi recebido o primeiro *beacon*; e o intervalo de tempo div em que o registo ainda é válido.

Sempre que um nó recebe um *beacon*, começa por analisar o endereço do emissor. O nó verifica se já possui um registo na tabela referente ao vizinho que o envia (caso contrário cria-o), copiando depois a informação do LGB eleito pelo vizinho (n_{LGB}) para o registo da tabela de *beacons*. O temporizador associado ao registo da tabela é depois (re)activado com o valor div sempre que é recebido um novo *beacon* de n_i . Após ter passado o tempo div sem se receber um *beacon* vindo do vizinho n_i , o registo é eliminado da tabela, indicando que a ligação lógica com o nó n_i foi quebrada. O valor pré-definido para o valor máximo de espera do temporizador poderá ser constante ou variável no tempo. Contudo, como o tempo de serviço das redes IEEE 802.11 é variável, a transmissão de um *beacon* pode sofrer grande atraso, e, caso o valor de div seja fixo e demasiado pequeno, pode suceder que o *beacon* seja recebido já fora do intervalo div e a ligação seja quebrada indevidamente.

A rede esquematizada na Figura 1 exemplifica a realização dos conceitos apresentados. A rede é composta por 6 nós, sendo os nós LGB representados por círculos pretos e os nós não LGB representados por circunferências. As linhas a tracejado representam as relações de vizinhança existentes, enquanto que as linhas a cheio definem os grupos de *broadcast* (GB). A Tabela I apresenta valores hipotéticos para a tabela de *beacons* do nó n_1 representado na Figura 1. A tabela apresenta três registos (linhas) relativos aos nós vizinhos n_2 , n_3 e n_4 . De notar que o nó n_1 se auto-elege como LGB, enquanto que os nós n_2 e n_3 elegem os LGB n_5 e n_6 , respectivamente. O nó n_1 possui uma ligação lógica com o vizinho n_2 há 43 múltiplos do período de *beacon* (div) sem que o temporizador associado ao registo da tabela desse vizinho tenha expirado. Desta forma, diz-se que a ligação mais estável do nó n_1 é a

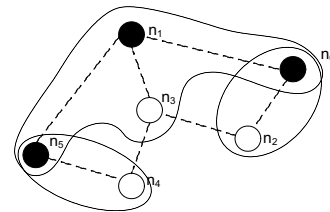


Figura 1. Rede *ad hoc* constituída por 6 nós (n_1 a n_6), onde existem 3 GB, e os nós n_1 , n_5 e n_6 são LGB.

Tabela I
EXEMPLO DO CONTEÚDO DA TABELA DE *beacons* DO NÓ REPRESENTADO NA FIGURA 1 NO INSTANTE 102.5S, SENDO $div = 10$ S.

n_i	S_{n,n_i}	n_{LGB}	t	div
43	59.2			
8	94.3			
2	100.1			

que mantém com o nó .

Diz-se que um nó possui uma ligação estável com um dos seus nós vizinhos no instante se onde é um limiar de estabilidade previamente definido, e representa o valor da estabilidade da ligação.

Um nó é estável se possui pelo menos um vizinho com o qual possui uma ligação estável. Um nó é denominado instável sempre que esta condição não se verifica. Por outras palavras, um nó é instável quando não possuir nenhuma ligação estável com qualquer dos seus nós vizinhos (a duração da ligação lógica expressa em múltiplos do período é sempre inferior a). O significado de estabilidade do nó permite caracterizar parcialmente a mobilidade do nó, pois permite identificar as relações de mobilidade relativa com os seus nós vizinhos. Note-se que, caso dois nós móveis possuam uma ligação estável, esta pode manter-se, mesmo quando os nós têm um valor de mobilidade elevada, desde que os dois nós estejam em posições espaciais que permitam a troca de *beacons* entre si.

Para a criação dos diferentes GB é necessário que todos os nós executem um algoritmo de eleição do seu próprio LGB. O algoritmo de eleição é distribuído e não necessita de qualquer outro tipo de troca de informação além da contida nos diferentes *beacons* enviados pelos nós vizinhos. Cada nó executa o algoritmo de eleição antes de enviar o *beacon*, a fim de enviar informação actualizada acerca do seu estado. No algoritmo, são utilizados os endereços dos nós, que se assume serem inteiros atribuídos univocamente a cada um dos nós. Antes de enviar um novo *beacon*, um nó elege o seu LGB, aplicando as seguintes regras:

- R1 - quando é instável, não elege nenhum LGB. Caso contrário, aplica as regras descritas a seguir;
- R2 - quando nenhum dos nós vizinhos de se encontra eleito como LGB de um GB, o nó elege como seu LGB um vizinho com o menor endereço de entre aqueles com o qual possui as ligações com maior valor de estabilidade de ligação;
- R3 - quando já se encontra eleito como LGB por um dos seus nós vizinhos, e todos os nós LGB vizinhos possuem um endereço superior ao seu, então o nó auto-elege-se como LGB;
- R4 - quando não é eleito por nenhum dos seus vizinhos, e existe pelo menos um nó vizinho que já se encontra eleito LGB, então o nó elege o nó como seu LGB. Quando existe mais do que um nó vizinho eleito LGB, elege o vizinho com o menor endereço.

É proposto o Algoritmo 1 para eleição do LGB. O algoritmo toma como parâmetros de entrada a informação contida na tabela de *beacons*. Na linha 1, a função "encontra_maximo_ _na_tabela_de_beacons()" devolve o maior valor de estabilidade de ligação actualizado a partir da tabela de *beacons* para ser aplicado na regra R2. Na linha 2, a variável "endereço" é iniciada com o maior inteiro possível. Caso seja estável com algum dos seus vizinhos, o algoritmo começa por criar uma lista ordenada de forma crescente com o endereço dos LGB eleitos pelos seus nós vizinhos (linhas

6 e 7), a qual pode também conter o nó , caso tenha sido eleito LGB por algum dos seus nós vizinhos (linhas 8 e 9). Se existirem nós vizinhos já eleitos LGB, elege como LGB o nó vizinho LGB que possui o menor endereço, o que é realizável porque os elementos retirados na lista na linha 10 encontram-se ordenados de forma crescente de endereços. Note-se, no entanto, que, caso algum vizinho tenha já eleito como LGB, o nó auto-elege-se LGB (linhas 17 a 18). Caso não existam nós vizinhos eleitos LGB, elege o seu nó vizinho com maior valor de estabilidade de ligação (linhas 20 a 24). Caso haja mais do que uma ligação com o maior valor de estabilidade, aplica-se um critério de desempate, elegendo como LGB o vizinho que possui o menor endereço. O algoritmo utiliza ainda a constante 'limiar_transiente=1' para considerar (no critério da linha 22) valores de estabilidade de ligações lógicas que ainda poderão valer durante o período de tempo .

```

parâmetros de entrada      :      , (      ),
                             (      ), (      )
parâmetros de saída        :
1  encontra_maximo_ _na_tabela_de_beacons()
2  endereço  MAX_INT
3  -1
4  limiar_transiente
5  if nó_estável then /* R1 - se este nó for estável */
6      for cada nó vizinho do
7          insere_em_lista_ordenada , lista_LGB /* menor
          endereço na cabeça da lista */
8      if é LGB then
9          insere_em_lista_ordenada( , lista_LGB)
10     for cada elemento lista_LGB do /* retira o elemento
na cabeça da lista */
11         for cada nó vizinho do
12             if e nó_estável then /* R4 - elege
um nó vizinho que já é LGB */
13                 |
14             if then /* acabou de eleger um LGB */
15                 break
16             if then /* R3 - auto-eleição */
17                 break
18             |
19             break
20         if = -1 then /* R2 - elege um nó vizinho sem
ser LGB */
21             for cada nó vizinho do
22                 if limiar_transiente e
23                     endereço then
24                         endereço
25                 |
26                 endereço
27             |
28             |
29             =

```

Algoritmo 1: Algoritmo de eleição do Líder de Grupo de Broadcast (LGB) do nó genérico .

A Figura 1 ilustra uma rede *ad hoc* móvel onde o Algoritmo 1 é executado. No instante inicial o nó é LGB, tendo sido eleito pelos nós , , e por ele próprio (auto-eleição). Esta eleição forma um GB composto pelos nós . Suponha-se que os nós e se movem para a vizinhança dos nós (,) e (,), criando noutro instante as ligações lógicas representadas na Figura 1. Quando as ligações entre os nós e e entre e se tornam estáveis, os nós e são eleitos LGB pelos nós e , respectivamente, criando dois GB. Tal como

se verifica no exemplo, o algoritmo origina uma árvore de grupos de *broadcast*, a qual é centrada no nó LGB possuindo o menor endereço. A Figura 2 apresenta a árvore de grupos de *broadcast* vista pelo nó n_1 , a qual é composta pelo nó raiz (n_1) e três ramos representados pelos nós n_5 , n_6 e n_3 . Cada ramo da árvore é constituído por um nó n_i , cujo LGB é o nó antecessor (n_{i-1}). As folhas da árvore (nós n_4 e n_2) representam os nós que não são eleitos LGB.

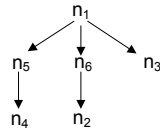


Figura 2. Árvore de grupos de *broadcast* do nó n_1 representado no exemplo da Figura 1.

O algoritmo origina um núcleo de nós (*backbone*) composto por nós LGB, o qual pode ser utilizado para inundar a rede. Os nós LGB podem estar ligados num núcleo que assegura a cobertura total da rede, tal como acontece entre os LGB da Figura 2. Neste caso, os nós LGB formam um conjunto de nós dominantes e conectados (CNDC) que assegura a cobertura total dos nós da rede. Ou seja, basta que os nós LGB transmitam o pedido de localização do recurso para que todos os nós da rede o recebam. Caso o conjunto de nós LGB existentes na rede não possua nós LGB a interligar dois GB, o CNDC terá de incluir pelo menos todos os nós LGB da rede e dois nós não LGB por cada nó LGB isolado existente na rede.

O desempenho do algoritmo de agrupamento depende da estabilidade da rede. O período de transmissão do *beacon* deve ser seleccionado de acordo com os valores de mobilidade dos nós. Caso uma grande percentagem dos nós seja estável, o algoritmo detecta-os, e os GB criados pelo algoritmo podem ser utilizados por outros algoritmos para diminuir a carga total da rede. Para manter a constituição dos grupos actualizada numa rede que apresenta mobilidade muito elevada, o algoritmo poderá exibir custos demasiado elevados: as rápidas alterações observadas na rede implicam um aumento da frequência de transmissão dos *beacons* para que o estado das ligações lógicas com os nós vizinhos possa ser identificado com o menor grau de incoerência. O aumento da frequência de transmissão dos *beacons* origina um aumento na carga da rede, a qual causa outros problemas, tais como o aumento das colisões entre tramas e a degradação de *throughput*. Consequentemente, é importante caracterizar a rede em termos da sua mobilidade máxima, pois a frequência de transmissão dos *beacons* deve ser adequada à velocidade máxima dos nós, por forma a que as ligações com os nós vizinhos sejam correctamente detectadas com valores de carga longe da situação de tempestade de *broadcast*.

III. OLSR-FCT: INTEGRAÇÃO DO ALGORITMO DE CONTROLO DE TOPOLOGIA NO PROTOCOLO OLSR

O protocolo de encaminhamento OLSR [2] não utiliza nenhuma informação acerca da mobilidade dos nós. Basicamente

a topologia da rede é conhecida através da difusão periódica de mensagens HELLO. Um determinado nó n_i recebe mensagens HELLO de todos os seus vizinhos n_j . É a partir das mensagens de HELLO enviadas por um nó n_j que o nó n_i sabe o conjunto dos nós vizinhos do vizinho n_j . O conjunto dos vizinhos dos vizinhos do nó n_i é representado por $2N_i$. Para garantir que todos os vizinhos dos vizinhos de n_i sabem da sua existência, o nó n_i elege um ou mais nós MPB a partir do conjunto $2N_i$, os quais representam o menor número de nós que deverão difundir as mensagens de topologia afim de cobrir todos os nós contidos em $2N_i$. Porém, num cenário de auto-estrada, o OLSR poderá originar um número elevado de nós MPB para garantir a cobertura total de $2N_i$, quando muitos dos nós contidos em $2N_i$ podem ter uma velocidade relativa muito elevada que não justifiquem serem cobertos por um nó MPB (dado que essa informação será inválida a curto prazo). Outro dos problemas do protocolo OLSR é o facto de poder eleger nós MPB que rapidamente deixam de ser vizinhos de n_i . Dessa forma, é preferível um nó eleger um vizinho seu como MPB com o qual mantenha maior estabilidade, pois a probabilidade de continuar conectado com ele é superior. Num cenário de auto-estrada, onde exista uma densidade de nós num dos sentidos com um valor capaz de cobrir toda a rede, é preferível um nó só eleger nós MPB no seu sentido pois a duração da conectividade com esse nó será superior à de um nó em sentido contrário.

Para integrar o mecanismo de controlo de topologia no protocolo OLSR foram adicionados à mensagem de HELLO os campos que caracterizam o *beacon*, nomeadamente:

- um campo para identificar se o nó emissor é um nó LGB;
- um campo que indica a associatividade máxima do nó (max_assoc).

Finalmente, o campo *willingness* já existente na mensagem de HELLO passa a valer 0 (*WILL_NEVER*) quando o nó emissor é instável ou o valor por omissão (3) quando o nó emissor é estável. O algoritmo não utiliza os vizinhos classificados de *WILL_NEVER* na realização dos caminhos. O algoritmo de controlo de topologia é executado em simultâneo com o protocolo OLSR sendo os nós MPBs eleitos de acordo com a informação obtida através do mesmo.

O algoritmo 2 sumariza os passos envolvidos na selecção dos nós MPB utilizando a informação fornecida pelo algoritmo de controlo de topologia. O algoritmo é aplicado ao nó n_i . A variável 'cobertura' é definida entre 0 e 1 e representa a percentagem actual de nós cobertos em $2N_i$ dado o conjunto de nós já escolhidos em $2N_i$. Nas linhas 3 a 8 são eleitos MPB os vizinhos de n_i que já tenham sido eleitos LGB e que sejam estáveis com o nó n_i . Nesta fase, caso o algoritmo de controlo de topologia seja devidamente parametrizado, os nós LGB estáveis com n_i representam um conjunto de nós que circulam no sentido de n_i e, dadas as suas características de mobilidade, deverão realizar a difusão da topologia. Para cada nó n_j eleito MPB é actualizada a percentagem de cobertura em $2N_i$ (linha 6) sendo esse nó retirado da lista de vizinhos (linha 7). No entanto, os nós MPB eleitos no critério anterior podem

não garantir a taxa de cobertura desejada. Caso se verifique este caso (linha 10), vão-se elegendo novos nós (linhas 11 a 13) utilizando como critério de eleição o maior valor de estabilidade de ligação (em caso de empate, é eleito o nó que consegue maior taxa de cobertura).

```

parâmetros de entrada      :
parâmetros de saída        : ,cobertura_requerida
                             : MPR
1  MPR
2  cobertura
3  for cada nó vizinho      do
4  if e é_LGB then /* R1 - se este nó for
   estável e LGB então deverá ser MPR */
5  MPR( ) MPR( )
6  actualiza(cobertura)
7  retira_da_lista( , )
8
9  = ordena_ _descendente( )
10 while cobertura < cobertura_requerida do
11   retira_da_lista( , )
12   MPR( ) MPR( )
13   actualiza(cobertura)
14

```

Algoritmo 2: Algoritmo proposto para eleição de nós MPR no protocolo OLSR.

O algoritmo de eleição de MPRs do protocolo OLSR é óptimo, na medida em que escolhe o menor número de nós em que têm de difundir a informação de forma a inundar todos os nós em . No entanto, a elevada mobilidade origina uma elevada necessidade de actualização da topologia, gerando uma elevada quantidade de tráfego que degrada o comportamento da rede. No algoritmo de selecção de MPRs, quando a cobertura requerida é 1 o número de MPRs é no melhor caso igual ao escolhido pelo algoritmo de selecção de MPRs utilizado no protocolo OLSR. É por isso que no algoritmo se utiliza normalmente uma percentagem de cobertura inferior a 100%, de forma a atenuar o impacto das múltiplas difusões efectuadas pelos nós MPR. Na próxima secção de avaliação do algoritmo estuda-se o comportamento para diversos valores de cobertura requerida.

Para que todos os nós vizinhos de que circulem no sentido contrário ao de sejam declarados instáveis, admite-se os nós possuem uma velocidade média de 75 Km/h. Nesta situação, admitindo que os nós possuem um raio de alcance de rádio de aproximadamente 1000 metros, um nó possui uma velocidade relativa de 150 Km/h face a um nó que se encontre em sentido contrário, o que equivale a estarem no alcance de rádio um do outro durante 48 segundos. Desta forma, parametrizando o período de HELLO () a 1 segundo, o parâmetro é parametrizado a 50, o que permite que um nó seja declarado instável face a outro, caso possua uma velocidade relativa superior a 144 Km/h.

IV. ANÁLISE DE DESEMPENHO

Para validar o algoritmo proposto utilizou-se o modelo de micro-mobilidade SUMO proposto em [5]. Neste modelo, os veículos seguem um veículo que circule à sua frente guardando uma determinada distância de segurança que é função da sua velocidade, aceleração e do tempo de reacção do condutor. Caso existam múltiplas faixas no mesmo sentido, o veículo da frente circula na faixa direita e os veículos que circulam

atrás podem ultrapassar utilizando outras faixas à esquerda. A velocidade do veículo da frente é limitada pelo valor definido para a sua velocidade máxima.

O cenário de mobilidade para análise de desempenho é constituído por um troço de auto-estrada com 10 Kms, onde existem três faixas de rodagem em cada um dos sentidos. Foram utilizadas 120 viaturas constituídas em três classes de tráfego diferentes. 60% das viaturas é da classe 1, caracterizadas por uma velocidade máxima de 100 Km/h, aceleração e desaceleração de 3.5 m/ . Das restantes, 25% alcançam uma velocidade máxima de 93.6 Km/h com aceleração e desaceleração 2.5 e 3.5 m/ enquanto que somente 15% alcança a velocidade máxima de 72 Km/h utilizando aceleração e desaceleração 1.5 e 2.0 m/ . Inicialmente são dispostos 30 veículos em cada sentido encontrando-se distribuídos pela auto-estrada de forma a observarem uma densidade de aproximadamente 5 vizinhos. Posteriormente entra um novo veículo na auto-estrada a cada 12 segundos (em média) e em cada sentido. A rede é simulada durante 1000 segundos.

Os movimentos dos veículos gerados pelo modelo SUMO foram depois integrados no simulador de redes ns-2 [6]. Utilizou-se a realização da norma IEEE 802.11 [7] disponibilizada no simulador como protocolo de nível físico e de sub-nível de acesso ao meio. Na norma foi usada a parametrização por omissão do simulador, embora tenham sido alterados o alcance de rádio para 1000 metros, e o tempo de expiração do *acknowledge* para 2.2 micro-segundos, por forma a garantir a transmissão das mensagens de *acknowledge* dentro do intervalo de *timeout*. Os ritmos de transmissão utilizados foram 11 Mbps para o ritmo de transmissão de dados (RTS/CTS) e 2 Mbps para o ritmo de transmissão básico.

Para simular o protocolo OLSR foi utilizada a realização do protocolo disponibilizada em [8]. O código disponibilizado em [8] também serviu de base à integração do algoritmo de topologia, tendo sido alterado o método de selecção dos nós MPR. Esta modificação ao protocolo OLSR é denominado OLSR-FCT. No algoritmo de controlo de topologia foram utilizados os parâmetros s e . No protocolo OLSR foram utilizados os períodos 1 e 2 segundos para a transmissão de mensagens de HELLO e de topologia, respectivamente.

O desempenho do protocolo OLSR com e sem as alterações propostas neste trabalho foi avaliado através da geração de aproximadamente 7300 pedidos de encaminhamento, os quais foram efectuados para nós de destino contidos na mesma faixa de rodagem e considerando somente os nós que ainda se encontravam activos (em movimento). Os pedidos de encaminhamento são registados obtendo-se estatísticas da percentagem de sucesso de resolução dos pedidos, bem como do tempo necessário à resolução (atraso do caminho).

Existem várias razões para que o pedido de encaminhamento falhe, as quais podem estar relacionadas com a inexistência de caminho devido à existência de grupos de nós desconectados, a estados de incoerência do algoritmo de encaminhamento, a um elevado nível de tráfego na rede (congestão), ou à existência de ciclos que originam perda

de pacotes por ter excedido o valor de TTL (*time-to-live*). Através da escolha de MPRs mais estáveis este trabalho pretende diminuir o número de pedidos de encaminhamento falhados devido à incoerência do algoritmo e da diminuição de situações de congestão observadas na rede.

A figura 3 apresenta a percentagem de sucesso dos pedidos de encaminhamento no cenário descrito nesta secção. A curva denominada OLSR apresenta a percentagem de sucesso utilizando o protocolo OLSR a qual foi obtida para uma única simulação replicada 10 vezes para reduzir o erro. Quanto à curva denominada OLSR-FCT, apresenta os resultados de sucesso relativamente ao protocolo OLSR integrando as alterações propostas. Nesta curva, o intervalo de confiança a 95% representado nos segmentos de recta verticais identificam também os valores de cobertura utilizados nas diferentes simulações (0.45, 0.6, 0.75, 0.8, 0.85, 0.9, 0.95 e 1). As modificações introduzidas no protocolo OLSR-FCT apresentam uma percentagem de sucesso de resolução dos pedidos de encaminhamento superior ao protocolo OLSR, apresentando-se a diferença mais elevada para o valor de cobertura 0.85, onde o protocolo OLSR-FCT exibe uma percentagem de sucesso de 61.2% face a 47.6% no caso do OLSR (um acréscimo relativo de 28.6%). Para valores de cobertura superiores a 0.85, a percentagem de sucesso começa a diminuir. Para estes valores, o elevado número de MPRs necessário para difundir a topologia começa a ser menos eficaz em termos da percentagem de sucesso.

A figura 4 apresenta o atraso dos caminhos resolvidos relativos ao cenário apresentado na figura 3. As simulações com o protocolo OLSR evidenciam um atraso médio do caminho de aproximadamente 0.099s. O intervalo de confiança é elevado devido ao elevado desvio padrão das medidas. Este facto é explicado pelo diferente comprimento dos caminhos pois o atraso do caminho é fortemente influenciado pelo número de nós que o constituem. Os valores do atraso no caso da alternativa OLSR-FCT são menores, sendo o seu valor médio sempre inferior (no pior caso o atraso diminui para cerca de metade).

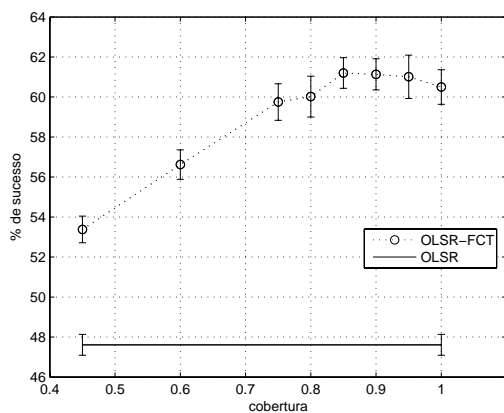


Figura 3. Percentagem de sucesso de resolução dos pedidos de encaminhamento (margem de erro a 95% de confiança).

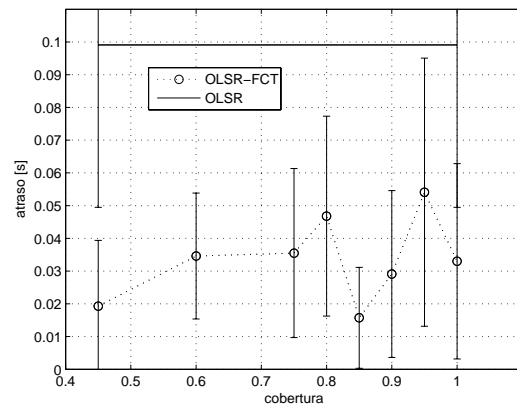


Figura 4. Atraso dos caminhos das simulações representadas na figura 3 (margem de erro a 95% de confiança).

V. CONCLUSÕES

Este trabalho apresenta um algoritmo para gestão de topologia em redes veiculares. O algoritmo é parametrizado para auto-estradas, sendo integrado no protocolo de encaminhamento OLSR, ao qual ainda se propõe uma nova metodologia para eleição dos nós que devem difundir a informação acerca da topologia. Os resultados de desempenho permitem concluir que se obtêm melhores resultados quer em termos da percentagem de sucesso da resolução dos pedidos de encaminhamento, quer em termos do atraso médio dos caminhos. As vantagens da proposta são ainda reforçadas pelo facto de não exigirem grandes modificações ao protocolo OLSR [2].

O trabalho futuro incluirá um estudo para determinação automática do valor de cobertura, bem como a possível utilização de alguns nós de faixas de sentido contrário para difusão da topologia quando a densidade de nós num sentido for insuficiente.

REFERÊNCIAS

- [1] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.
- [2] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol. *IETF Internet Draft*, <http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-10.txt>, 2003.
- [3] Rodolfo Oliveira, Luis Bernardo, and Paulo Pinto. The Influence of Broadcast Traffic on IEEE 802.11 DCF Networks. *Elsevier Computer Communications*, 32(2):439–452, 2009.
- [4] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.
- [5] Centre for Applied Informatics (ZAIK) and Institute of Transport Research at the German Aerospace Centre. SUMO - Simulation of Urban Mobility. Software Package retrieved from <http://sumo.sourceforge.net>, 2009.
- [6] Information Sciences Institute. NS-2 network simulator (version 2.31). Software Package retrieved from <http://www.isi.edu/nsnam/ns/>, 2007.
- [7] ANSI/IEEE 802.11 Standard. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- [8] Francisco J. Ros. UM-OLSR. Software Package retrieved from <http://masimum.inf.um.es/um-olsr/html/>, 2009.

Improving Routing Performance in High Mobility and High Density *ad hoc* Vehicular Networks

Rodolfo Oliveira, Luis Bernardo, Miguel Luis, Paulo Pinto
FCT-UNL, Universidade Nova de Lisboa, Portugal
UNINOVA, Monte da Caparica, Portugal

Abstract—In *ad hoc* networks the broadcast nature of the radio channel poses a unique challenge because the wireless links have time-varying characteristics in terms of link capacity and link-error probability. In mobile networks, particularly in vehicular *ad hoc* networks (VANETs), the topology is highly dynamic due to the movement of the nodes, hence an on-going session suffers frequent path breaks.

In this paper we present a method that uses the available knowledge about the network's topology to improve the routing protocol's performance through decreasing the probability of path breaks. We propose a scheme to identify long duration links in VANETs, which are preferentially used for routing. This scheme is easily integrated in the existent routing protocols. We describe how to integrate it in the Optimized Link-State Routing Protocol¹. Finally, we evaluate the performance of our method with the original protocol. Simulation results show that our method exhibits better end-to-end path delay (almost one magnitude order lower) and packet delivery ratio (between 25% and 38% higher) than the original protocol. This observation is even more evident when the node's density increases.

Keywords: Topology Control, Routing Protocols, Vehicular *ad hoc* Networks.

I. INTRODUCTION

Emerging vehicular networks are rapidly becoming a reality. Nowadays, several organizations are supporting standardization activities that will enable a variety of applications such as safety, traffic efficiency, and infotainment. Vehicular *ad hoc* networks (VANETs) share some common features with the traditional mobile *ad hoc* networks (MANETs), namely in terms of self-organization of the nodes. But they also differ in some issues: in VANETs the level of node's mobility is generally higher, the mobility is constrained by the roads and in terms of energy the nodes are not so constrained as in MANETs. Due to the fast change of the topology, VANETs demand for routing protocols focused on decreasing the number of path breaks.

The routing protocols that have been proposed for Mobile *Ad Hoc* Networks can be classified into three basic groups [1]: unicast topology-based, unicast position-based or group-based multicast and broadcast.

In topology-based protocols the nodes need to store routing tables or routes that depend on the topology. This class of protocols include the well known *Ad hoc* On-Demand Distance Vector Routing (AODV) [2], Optimized Link State

Routing (OLSR) [3] and others (see [4], [5]). Traditionally these protocols were proposed for MANETs, where the nodes are assumed to have *moderate* mobility. This assumption allows these protocols to establish end-to-end paths that are valid for a reasonable amount of time and only occasionally need repairs. Therefore, these conditions are only valid in some vehicular scenarios, where the maximum speed of the nodes is strongly restricted. For high mobility scenarios, such as highways, the nodes exhibit unique characteristics [6] and the routing protocols used for MANETs do not perform well on VANETs [1]. For high mobility scenarios, topology-based protocols also pose other challenges related with the topology changes: usually they continuously maintain up-to-date routes for valid destinations and require periodic updates to reflect network topology changes. This requirement can lead to high bandwidth consumption, which can be alleviated by some optimization processes, such as the MultiPoint Relay (MPR) scheme used in [3].

In unicast position-based routing protocols [7], the nodes do not need to store any route or routing table to the destination. Instead, the nodes use the location of their neighbors and the location of the destination node to determine the neighbor that forwards the packet. Therefore, these routing schemes require information about the position of the nodes, which is a drawback when the positioning system fails (e.g. the GPS receivers can lose the signal inside tunnels).

In this work we approach the topology-based routing class, which does not require positioning systems. We intend to use it in a high mobility scenario such as highways, to provide the deployment of comfort applications such as onboard games and video/music file sharing. Based on the fact that this class of protocols, namely those where the nodes store routing tables [2] [4] [3], use periodical broadcast of Hello messages to discover its neighborhood, we present a scheme to detect long duration links between vehicles. If properly used by the routing protocol, long-duration links are supposed to decrease the routing instability, decreasing the number of routing path breaks. The neighbors with which a node maintains long duration links are also identified and grouped. The groups are used to decrease the amount of broadcasts required to disseminate network topology changes.

Our approach is easily integrated in the existent routing protocols. We describe how to integrate it in the Optimized Link-State Routing Protocol [3] and we evaluate the performance of our method. Simulation results show that our method exhibits better end-to-end path delay (almost one magnitude order lower) and path availability for each destination (between 25% and 38% higher) when compared to the OLSR original

¹The source code of our proposal, entitled OLSR-FCT, was written for the network simulator ns-2.33 and is available to download at <http://tele1.dee.fct.unl.pt/people/rado/html/downloads.html>, allowing the community to evaluate their own scenarios and compare it with other protocols.

protocol. This observation is even more evident when the node's density increases.

The rest of the paper is organized as follows. Section II motivates and describes the problem approached in this work. In Section III we describe how the long duration links are detected, and we introduce the algorithm that groups the neighbors which maintain long duration links. This section ends with an example of how our proposals can be incorporated in the Optimized Link-State Routing Protocol. Section IV presents the experimental results. Finally, some concluding remarks are given in section V.

II. MOTIVATION AND PROBLEM DESCRIPTION

A. Motivation

The work presented in this paper is motivated by results obtained through simulations. We have simulated a VANET with 120 vehicles on a segment of a straight line highway with 3 lanes and 10 kms long. The simulation started with 30 vehicles moving on each side of the highway. During the simulation we launch more 30 vehicles on each side of the highway to maintain a constant density of nodes in the network. Each vehicle generates 0.5 packets per second and has, on average, 6 vehicles in its radio range². The packets are randomly destined to the vehicles moving in the same way. The OLSR routing protocol was used.

In the first simulation, we evaluate the case when the multihop path only uses vehicles moving in the same way. Therefore, the radios of the nodes moving in one side of the highway were turned off. The simulated results, presented in table I, indicate that 68.8% of the packets were successfully delivered in an average time of 66.9ms. In the second simulation we repeated the first simulation setup, except that all vehicle's radios were turned on. Thus, the multihop paths can use the nodes moving in the opposite way. In this case, the routing performance diminishes almost 32% in terms of packet delivery ratio (from 68.8% in the first simulation to 47.1% in the second one). The same behavior is observed for the average end-to-end delay. It increases approximately 48% (from 66.9 ms to 99.1ms). These results indicate that the use of the vehicles in the opposite way can severely damage the routing performance in terms of both packet delivery ratio and end-to-end delay. These observations motivate this work, which aims to improve the routing performance for the scenario considered in the second simulation.

TABLE I

OLSR RESULTS FOR MULTIHOP PATHS THAT ONLY USE NODES MOVING IN THE SAME WAY (SINGLE WAY) OR IN BOTH WAYS.

	single way	both ways
packet delivery ratio	68.8%	47.1%
average end-to-end delay	66.9 ms	99.1 ms

B. Problem Analysis

We start to consider the mobility scenario shown in Figure 1, where the vehicles 1 to 4 are moving at velocities

to , respectively. In this analysis we adopt the following assumptions: two nodes are length unities far away from each other; the radio communication range of each node is expressed by ; a link is detected and subsequently sensed if

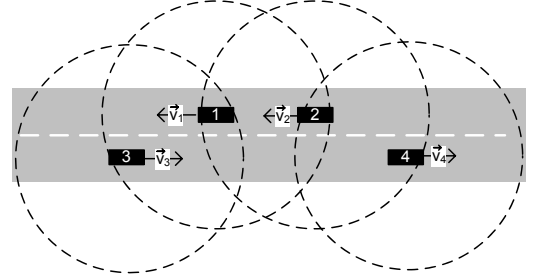


Fig. 1. Mobility scenario considered in the analysis.

Representing the velocity of the nodes and by the vectors and and being , we differentiate two cases:

- in this case the link between the nodes will remain active while this condition holds true (e.g.: and depicted in Figure 1);

- this condition imposes that the link will be broken after some time (e.g.: and or and shown in Figure 1);

Representing and in polar coordinates and , with and , we represent the relative velocity of the nodes by

(1)

and its absolute value is defined as

(2)

The relative velocity is a function that depends on the random variables e , which are mutually independents. Considering a random variable that expresses the relative velocity , the expected relative velocity value is defined as

(3)

As the random variables in (3) are independent, the condition is valid and yields

(4)

Assuming that at instant two nodes and form a link and, considering that the node moves with velocity relative to node , the link will be considered broken if after some time. Assuming that nodes do not change their velocity between the interval (,), which is a good

²Section IV gives more details about the simulated scenario.

approximation as $\frac{d}{2}$, the nodes will maintain a link active if during the interval (t_1, t_2) the distance between the nodes never exceeds $\frac{d}{2}$.

The probability of an existing link at time t remaining active in the time t_1 is related with the spacial intersection of the covered areas at instants t_1 and t_2 (the space covered at both instants), which is represented by the shaded area in Figure 2.

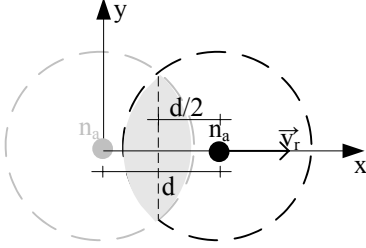


Fig. 2. Position of the node n_b in the time instants t_1 and t_2 after moved length units after the instant t_1 .

Knowing that the radio covering circumference of the node at instant t_1 is given by

$$C_1 = \pi d^2 \quad (5)$$

the overlapped area in the instant t_2 (represented by C_2 and illustrated by the shaded area in the Figure 2) is a function of the distance d travelled by the node with velocity v in the interval (t_1, t_2) , and is given by

$$C_2 = \pi \left(\frac{d}{2} \right)^2 \quad (6)$$

Now we consider the case when Hello messages are broadcasted every T seconds to discover and/or maintain an active link. The distance travelled by the node n_b relative to the node n_a during the period T is given by $d = vT$. Therefore, the probability of the link remains active during T periods is given by

$$P = \frac{C_2}{C_1} \quad (7)$$

When the nodes n_a and n_b are moving in the same way, we have $v_r = v$ or $v_r = -v$. In this case, the expected relative velocity value yields

$$v_r = v \quad (8)$$

When the nodes n_a and n_b are moving in the opposite way, we have $v_r = 2v$ or $v_r = -2v$. In this case, the expected relative velocity value yields

$$v_r = 2v \quad (9)$$

Because $\frac{d}{2}$ is a decreasing function in t , the link between the nodes n_a and n_b has an higher probability P of remaining active when the nodes are moving in the same way. This conclusion should be adopted by the routing protocol: the multihop path created by the routing protocol should preferentially use links between nodes moving in the same way, because the lower probability of link breaks between these nodes will decrease the probability of path breaks.

III. ROUTING IMPROVEMENTS

A. Long Links Detection

Based on the description previously presented, this subsection introduces a solution to detect the links formed by two nodes moving in the same direction.

In topology-based routing algorithms the links between the nodes are discovered and maintained through periodical Hello packets exchange. The duration of the link is characterized by the number of Hello packets uninterruptedly received. We start to define the notion of node's **logical neighborhood**: the set of logical neighbors N_L is the set of 1-hop nodes from which the node n_a receives Hello packets.

In the time instant t_1 , when the node n_a firstly receives an Hello packet from it's neighbor node n_b , an unidirectional logical link is created between the nodes. The duration of the logical link can be quantified by its **stability** value: the stability S measures the duration of the logical link between the nodes n_a and n_b . S is computed by the node n_a at instant t_2 by applying the following expression³

$$S = \frac{t_2 - t_1}{T} \quad (10)$$

Each node maintains its own table of logical neighbors to detect the links created in the same moving direction. Each line of the table represents one logical neighbor and contains information about n_b 's neighbors address ($addr$), their stability value (S), the time instant when the first Hello packet was received by n_a (t_1) and a time interval (T) during while the information described in the line is valid. Table II represents an hypothetic table of logical neighbors of node n_a represented in the MANET of Figure 3.

TABLE II
TABLE OF LOGICAL NEIGHBORS OF NODE n_a (ILLUSTRATED IN FIGURE 3)
AT THE INSTANT t_1 AND CONSIDERING T .

Neighbors address	Stability value	Time instant	Time interval
65	37.2		
30	72.3		
2	100.1		

The logical links between the nodes moving at the same direction are identified by each node through the observation of each link stability. A logical link is said to be stable if it lasts longer than a given S_{th} value:

$$S > S_{th} \quad (11)$$

By (6) and (7), a link created by two nodes moving in opposite directions presents a null probability of remaining

³The symbol div is used for integer division.

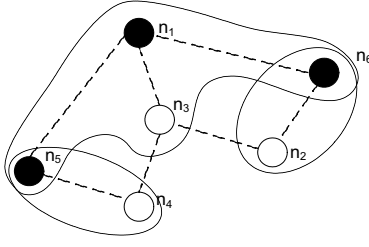


Fig. 3. MANET formed by 6 nodes.

active when \dots . Thus, for a link created by two nodes moving in opposite directions, the condition \dots only holds when \dots opposite way \dots . Therefore, when the condition \dots

$$\dots \quad (12)$$

holds, the stable links identified by the condition (11) represent the links maintained by the nodes moving in the same direction, because the links between vehicles moving in opposite directions never reach a stability \dots greater than the number of periods given by \dots opposite way \dots . In the section IV we exemplify how to achieve \dots opposite way \dots and for a real scenario.

B. Broadcast Leader Selection

The neighbors with which a node maintains stable links (stable neighbors) are more suitable to advertise network topology changes, because these nodes sense less link breaks. To decrease the amount of broadcasts needed to flood the network with the topology changes, we propose an algorithm that groups stable neighbors into 1-hop radius groups. Each node selects a single Broadcast Group Leader (BGL). BGLs are preferentially used to broadcast the network topology changes. We denote \dots as the BGL node selected by the node \dots .

In the network depicted in Figure 3, the node n_1 selects n_1 as its own BGL, while node n_2 selects the node n_3 . The remaining nodes n_3 and n_4 select the node n_5 as their BGL. Thus 3 different broadcast groups (BGs) are formed, represented by the sets $\{n_1, n_3, n_5\}$, $\{n_2, n_4, n_6\}$ and $\{n_3, n_4, n_5\}$. Now only the BGL nodes n_1 , n_2 and n_5 are requested to broadcast in order to flood the entire network.

Considering a generic node n_i , and admitting that n_i knows the BGL nodes chosen by its neighbors ($n_{j_1}, n_{j_2}, \dots, n_{j_k}$ - which are transmitted in the Hello packet), n_i selects its own BGL by applying the Algorithm 1. The algorithm rules R1-R4 are summarized as follows:

- R1 - when n_i is unstable (meaning that n_i does not have stable neighbors) it does not select any BGL;
- R2 - when none of n_i 's neighbors ($n_{j_1}, n_{j_2}, \dots, n_{j_k}$) had previously selected a BGL, n_i selects the neighbor having the smaller address from the set of the neighbors with which n_i has the biggest stability value;
- R3 - n_i selects itself as a BGL when n_i is already a BGL node (previously selected by a neighbor) and the neighbors's BGL have higher addresses than n_i ;

```

Input : ( ), ( ), ( ), ( )
Output :

1 return_max_ from_beacon_table()
2 address MAX_INT
3 -1
4 transient_threshold
5 if stable_node then /* R1 - if this node is stable */
6   for each neighbor do
7     insert_sorted( , list_BGL) /* lower addresses in
8     the head of the list */
9   if is_BGL then
10    insert_sorted( , list_BGL)
11  for each list_BGL do /* is removed from the
12  head of the list */
13    for each neighbor do
14      if and stable_node then /* R4 -
15      select a neighbor that already is BGL */
16      if then /* BGL already selected */
17        break
18      if then /* R3 - auto-selection */
19        break
20  if = -1 then /* R2 - its neighbor becomes a new
21  BGL */
22  for each neighbor do
23    if address then
24      address
25    address
26  address
27  address
28  address
29  =

```

Algorithm 1: Algorithm used by the generic node to select its Broadcast Group Leader.

- R4 - when n_i is not selected BGL by its neighbors and there exists at least one neighbor n_j that is already a BGL, selects the node n_j as its own BGL; ties are broken by choosing the smaller address neighbor;

The first BGL node selected in the network is justified by the application of the rule R2. The rule R4 is defined to merge several BGs. The rule R3 is also used to merge several BGs in the special case when n_i must select itself as a BGL.

C. Integrating the topology information in OLSR Routing Protocol

This subsection exemplifies how the stable links and the Broadcast Groups are integrated in the Optimized Link-State Routing Protocol [3]. The OLSR routing protocol uses Multipoint relaying. Multipoint relaying is a technique to reduce the number of redundant re-transmissions while diffusing a broadcast message in the network [10]. Basically, a node chooses a set of Multipoint Relay nodes (MPRs). MPRs are chosen as the minimum set of n_i 's 1-hop neighbors that cover all its neighbors 2-hops away. Thus MPR nodes guarantees 2-hops full coverage.

We use the BGL nodes in the MPR's selection algorithm, proposing a new algorithm to compute the MPRs (Algorithm 2). For a given node n_i , the algorithm needs to know about its neighbors ($n_{j_1}, n_{j_2}, \dots, n_{j_k}$), their stability ($n_{j_1}, n_{j_2}, \dots, n_{j_k}$), their BGLs ($n_{j_1}, n_{j_2}, \dots, n_{j_k}$) and the required coverage. OLSR uses the smallest set of MPRs that guarantees the full coverage 2-hops away. Because our algorithm does not guarantees the

smallest set of MPRs to cover the neighbors 2-hops away, it may introduce undesired control traffic overhead. This is the main reason to design our algorithm with an input parameter named `required_coverage`, that allows us to characterize our algorithm's behavior for different amounts of coverage.

```

Input      : required_coverage
Output    : MPR

1 MPR
2 coverage
3 for each_neighbor do
4   if stable and BGL then /* R1 - if is
5     MPR( ) MPR( )
6     update(coverage)
7     remove_from_list( , )
8
9   = sort_by_descendent_ ( )
10 while coverage required_coverage do
11   remove_from_list( , )
12   MPR( ) MPR( )
13   update(coverage)
14

```

Algorithm 2: New algorithm to select OLSR's MPR nodes used by the OLSR.

The algorithm is summarized as follows. 1-hop neighbors of that have been selected BGLs and have stable links with are selected MPRs (lines 3 to 8). If the number of MPRs are not guaranteeing the required coverage 2-hops away (line 10), the algorithm selects the MPR nodes with which has the highest stability values.

Regarding the routing table's computation, the OLSR uses a field named Willingness that specifies the willingness of a node to carry and forward traffic for other nodes. Only 1-hop neighbors with Willingness different of `WILL_NEVER` are used to forward packets. In our proposal, we use the OLSR's Willingness mechanism to prohibit's unstable neighbors from forwarding packets. We assign the willingness of's 1-hop neighbors that do not meet the condition stated in (12) to `WILL_NEVER`.

IV. PERFORMANCE EVALUATION

A. Mobility Model

The VANETs simulated in our evaluation scenario were obtained using the Trans tool [11], which integrates the SUMO traffic simulator [8]. We have simulated a segment of a straight line highway with 3 lanes in each direction. The simulations start with the vehicles moving in both sides of the highway. During the simulation we launch more vehicles to maintain a constant density of nodes in the network. The highway segment is 10 kms long, which limits the minimum number of the network hops to 10. We defined three different classes of vehicles, which are described in the table III. 60% of the vehicles belong to the class , which represents medium size cars. The vehicles of class represents 25% of the highway traffic. Finally we define 15% of vehicles of class , which represents long sized vehicles such as trucks. Regarding the vehicle's density, we defined 3 different scenarios, described in the table III. All vehicles are assumed to have a radio range of 1000 meters. The scenario Scen was the same used to obtain the motivation results presented in the table I.

TABLE III
CLASSES OF TRAFFIC CONSIDERED IN THE SIMULATIONS.

	vehicle's length (m)	(m/s)	acceleration (m/s ²)	deceleration (m/s ²)	%
class	4	27.8	3.6	3.6	60
class	5	26.0	2.5	3.0	25
class	8	20	1.5	2.0	15

TABLE IV
VEHICLE'S DENSITIES CONSIDERED IN THE SIMULATIONS.

	number of vehicles	average number of neighbors	simulation time (s)
Scen	120	6.0	830
Scen	160	8.0	915
Scen	200	10.0	886

B. Simulation Description

The simulations compare the performance of our improvements with the OLSR routing protocol. We used the simulator ns-2 [9] configured with the standard IEEE 802.11⁴ and the OLSR protocol implementation available from [12]. Our proposals were integrated in the OLSR implementation, being the modified protocol designated OLSR-FCT.

Vehicles moving in the same left-to-right direction generate 0.5 packets per second, which are randomly destined to the nodes in the same direction. The vehicles moving from right-to-left do not generate packets but are able to forward them. The number of packets generated on each density scenario were 7388, 10102 and 13220⁵ (from Scen to Scen , respectively). We simulated 6 different situations: 3 density scenarios previously described using the OLSR protocol and the same scenarios using our OLSR-FCT protocol.

To parameterize we assumed a rough approximation for the three classes of traffic, as being normally distributed with and average and 0.5m/s of standard deviation. For this case the relative velocity between two vehicles moving in opposite directions yields opposite way m/s, and the relative velocity between two vehicles moving in same direction (same way) yields approximately 2m/s. Choosing the Hello transmission frequency () to 1Hz, the minimum value that guarantees unstable links formed by opposite moving vehicles is 38.19 (given by (12) and considering m). Therefore, we chosen = 50 to have a security margin of approximately 12 periods. This margin avoids the vehicles moving at lower speeds to be erroneously declared as moving in the same direction. This parameterizations indicates that the links of the vehicles moving in the opposite direction never last longer than 50s. Moreover, when the links of the vehicles moving in the same direction last for 50s, their probability of still being active is 96.8% ().

C. Experimental Results

The OLSR protocol was firstly simulated for the 3 scenarios described in table IV. The path delivery ratio and the average

⁴11 Mbps and 2 Mbps were used to transmit unicast and broadcast traffic, respectively.

⁵Note that a vehicle only generates packets when it is moving.

end-to-end delay were obtained within 95% of confidence interval and are shown in table V. The results indicate that the packet delivery ratio is approximately 50% in the 3 scenarios and the end-to-end increases with the vehicle's density.

TABLE V
OLSR EXPERIMENTAL RESULTS.

	packet delivery ratio (%)	end-to-end delay (ms)
Scen	47.13%	99.10
Scen	53.94%	221.37
Scen	52.75%	301.59

The results obtained with our OLSR-FCT proposal are shown in the Figures 4 and 5. The OLSR-FCT was tested for different values of coverage (45%, 60%, 85% and 100%, and the coverage means the same as defined in the Algorithm2). The results obtained with the OLSR-FCT protocol present a better performance when full coverage is used (100%). For full coverage the number of MPRs increase and, consequently, more topology traffic overhead is generated. This fact increases the end-to-end delay (Figure 5 shows higher average end-to-end delays when the coverage is 100%). The results plotted in Figure 4 show that the OLSR-FCT protocol exhibits better packet delivery ratio for higher node densities.

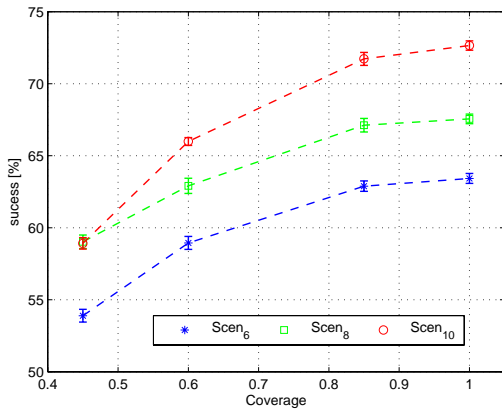


Fig. 4. Average packet delivery ratio (95% confidence intervals represented by vertical bars).

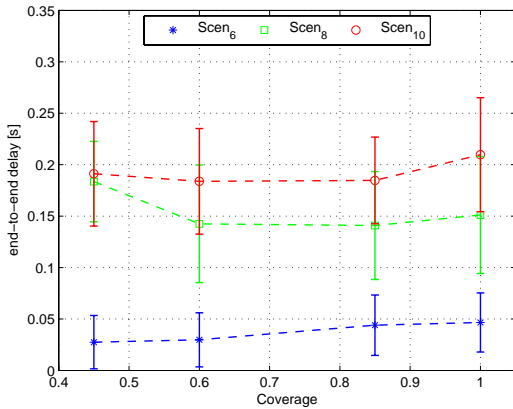


Fig. 5. Average end-to-end delay (95% confidence intervals represented by vertical bars).

Comparing the results presented in the motivation (obtained

with the original OLSR protocol) we conclude that the OLSR-FCT protocol improves the packet delivery ratio (from 47.1% to 66.9%) and almost achieves the same ratio as in the case when the radios of the vehicles in one direction were turned off (68.8%). In terms of end-to-end delay, the OLSR-FCT decreases the delay from 99.1ms to 42.2ms, being even shorter than the delay measured for the case when the radios of the vehicles in one direction were turned off (66.9ms). This fact is due to the use of long-duration links and control traffic decrease implemented by the OLSR-FCT protocol.

Finally, we present the gains obtained with OLSR-FCT in the table VI.

TABLE VI
OLSR-FCT EXPERIMENTAL GAINS WHEN 100% OF COVERAGE IS CONSIDERED.

	packet delivery ratio gain (%)	end-to-end delay gain (%)
Scen	33.2%	52.94
Scen	26.19%	31.76
Scen	37.72%	30.48

V. CONCLUSIONS

In this paper we present a method that uses the available knowledge about the network's topology to improve the routing protocol's performance through decreasing the probability of path breaks.

We integrate our improvements in the OLSR routing protocol. The performance results explicitly confirms that our proposal outperforms the original protocol, recommending its use for high mobility and high density scenarios.

REFERENCES

- [1] Jasmine Chennikara-Varghese, Wai Chen, Onur Altintas, and Shengwei Cai. Survey of routing protocols for inter-vehicle communications. In *Mobile and Ubiquitous Systems: Networking and Services, 2006 Third Annual International Conference on*, pages 1–5, July 2006.
- [2] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 25–26 Feb 1999.
- [3] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol. *IETF Internet Draft*, <http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-10.txt>, 2003.
- [4] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.
- [5] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
- [6] J.J. Blum, A. Eskandarian, and L.J. Hoffman. Challenges of intervehicle ad hoc networks. *Intelligent Transportation Systems, IEEE Transactions on*, 5(4):347–351, Dec. 2004.
- [7] M. Mauve, A. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *Network, IEEE*, 15(6):30–39, Nov/Dec 2001.
- [8] Centre for Applied Informatics (ZAIK) and Institute of Transport Research at the German Aerospace Centre. SUMO - Simulation of Urban Mobility. Software Package retrieved from <http://sumo.sourceforge.net>, 2009.
- [9] Information Sciences Institute. NS-2 network simulator (version 2.31). Software Package retrieved from <http://www.isi.edu/nsnam/ns/>, 2007.
- [10] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. *Hawaii International Conference on System Sciences*, 9:298, 2002.
- [11] TraNS. - open source tool for realistic simulations of VANET applications. Software Package retrieved from <http://trans.epfl.ch/>, 2009.
- [12] Francisco J. Ros. *UM-OLSR. Software Package* retrieved from <http://masimum.inf.um.es/um-olsr/html/>, 2009.

Bibliografia

- [Bar04] Rimón Barr. *An efficient, unifying approach to simulation using virtual machines*. PhD thesis, Ithaca, NY, USA, 2004. Adviser-Haas, Zygmunt J.
- [BE05] J.J. Blum and A. Eskandarian. Adaptive space division multiplexing: an improved link layer protocol for inter-vehicle communications. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 455–460, Setembro 2005.
- [BEH03] J. Blum, A. Eskandarian, and L. Hoffman. Mobility management in ivc networks. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 150–155, Junho 2003.
- [Bou04] Azzedine Boukerche. Performance evaluation of routing protocols for ad hoc wireless networks. *Mob. Netw. Appl.*, 9(4):333–342, 2004.
- [BSH00] L. Briesemeister, L. Schafers, and G. Hommel. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 522–527, 2000.
- [Bur] U.S. Census Bureau. Topologically integrated geographic encoding and referencing. <http://www.census.gov/geo/www/tiger/>.
- [CAR] CAR 2 CAR Communication Consortium.
- [CB05] David R. Choffnes and Fabián E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *VANET '05: Proceedings of*

- the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 69–78, New York, NY, USA, 2005. ACM.
- [CBD02] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.
- [CG98] Tsu-Wei Chen and M. Gerla. Global state routing: a new routing scheme for ad-hoc wireless networks. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 1, pages 171–175 vol.1, Junho 1998.
- [DB97] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Communications, 1997. ICC 97 Montreal, 'Towards the Knowledge Millennium'. 1997 IEEE International Conference on*, volume 1, pages 376–380 vol.1, Junho 1997.
- [DDB05] M. Duresi, A. Duresi, and L. Barolli. Emergency broadcast protocol for inter-vehicle communications. In *Parallel and Distributed Systems, 2005. Proceedings. 11th International Conference on*, volume 2, pages 402–406, Julho 2005.
- [FHFB07] Marco Fiore, Jérôme Haerri, Fethi Filali, and Christian Bonnet. Understanding vehicular mobility in network simulation. In *MoVeNet 2007, 1st IEEE international Workshop on Mobile Vehicular Networks, in conjunction with IEEE MASS 2007, October 8, 2007, Pisa, Italy*, Outubro 2007.
- [FHR04] Michael Feeley, Norman Hutchinson, and Suprio Ray. Realistic mobility for mobile ad hoc network simulation. In *Lecture Notes in Computer Science (LNCS)*, volume 3158, pages 324–329, 2004.
- [FMH⁺03] Holger Füssler, Martin Mauve, Hannes Hartenstein, Michael Käsemann, and Dieter Vollmer. Mobicom poster: location-based routing for vehicular ad-hoc networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(1):47–49, 2003.

- [HFB08] Jérôme Haerri, Fethi Filali, and Christian Bonnet. Mobility models for vehicular ad hoc networks: a survey and taxonomy. *Accepted for IEEE Communications Surveys and Tutorials (epublication)*, 2008.
- [HFBF06] J. Härrri, F. Filali, C. Bonnet, and Marco Fiore. Vanetmobisim: generating realistic mobility patterns for vanets. In *VANET '06: Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 96–97, New York, NY, USA, 2006. ACM.
- [Inf07] Information Sciences Institute. NS-2 network simulator (version 2.31). Software Package retrieved from <http://www.isi.edu/nsnam/ns/>, 2007.
- [JBW05] S. Jaap, M. Bechler, and L. Wolf. Evaluation of Routing Protocols for Vehicular Ad Hoc Networks in City Traffic Scenarios. In *Proceedings of the 5th International Conference on Intelligent Transportation Systems Telecommunications (ITST), Brest, France, June, 2005*.
- [JM96] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.
- [JMC⁺01] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. *Technology for the 21st Century, Proceedings of IEEE International Multi Topic Conference on*, pages 62–68, 2001.
- [KHWR02] Daniel Krajzewicz, Georg Hertkorn, Peter Wagner, and Christian Rössel. Sumo (simulation of urban mobility), an open-source traffic simulation. In *4th Middle East Symposium on Simulation and Modelling (MESM2002)*, pages 183–187, 2002.
- [KK00] Brad Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.

- [KML07] F. K. Karnadi, Zhi H. Mo, and Kun-Chan Lan. Rapid generation of realistic mobility models for vanet. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2506–2511, 2007.
- [KR09] Daniel Krajzewicz and Christian Rössel. *SUMO - Simulation of Urban Mobility - User Documentation*, 2009.
- [Kra98] Stefan Krauß. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. PhD thesis, 1998.
- [KSJ08] Maria Kihl, Mihail L. Sichitiu, and Harshvardhan P. Joshi. Design and evaluation of two geocast protocols for vehicular ad-hoc networks. volume 2, 2008.
- [LAP⁺01] D. Lee, R. Attias, A. Puri, R. Sengupta, S. Tripakis, and P. Varaiya. A wireless token ring protocol for intelligent transportation systems. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 1152–1157, 2001.
- [LG97] C.R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *Selected Areas in Communications, IEEE Journal on*, 15(7):1265–1275, Setembro 1997.
- [LLL⁺04] Genping Liu, Genping Liu, Bu-Sung Lee, Boon-Chong Seet, Chuan-Heng Foh, and Keok-Kee Lee. A routing strategy for metropolis vehicular communications. In *In Proc. International Conference on Information Networking*, pages 533–542, 2004.
- [LM07] Ilias Leontiadis and Cecilia Mascolo. Geopps: Geographical opportunistic routing for vehicular networks. In *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, pages 1–6, Junho 2007.
- [LOBP09] M. Luís, R. Oliveira, L. Bernardo, and P. Pinto. Controlo de topologia de redes ad hoc veiculares em cenários de auto-estradas. In *CRC 2009: 9ª Conferência sobre Redes de Computadores*, Outubro 2009.

- [LTP05] Pei Liu, Zhifeng Tao, and S. Panwar. A cooperative mac protocol for wireless local area networks. In *Communications, 2005. ICC 2005. 2005 IEEE International Conference on*, volume 5, pages 2962–2968 Vol. 5, Maio 2005.
- [LW07] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *Vehicular Technology Magazine, IEEE*, 2:12–22, Junho 2007.
- [MGLA96] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mob. Netw. Appl.*, 1(2):183–197, 1996.
- [MRR80] John M. McQuillan, Ira Richer, and Eric C. Rose. The new routing algorithm for the arpanet. volume 28, 1980.
- [MWL03] M. Bechler, W.J. Franz, and L. Wolf. Mobile internet access in fleetnet. In *In 13. Fachtagung Kommunikation in Verteilten Systemen, Kivs 2003*, Fevereiro 2003.
- [NAG04] Vinod Namboodiri, Manish Agarwal, and Lixin Gao. A study on the feasibility of mobile gateways for vehicular ad-hoc networks. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 66–75, New York, NY, USA, 2004. ACM.
- [NCG05] Sai Shankar N, Chun-Ting Chou, and M. Ghosh. Cooperative communication mac (cmac) - a new mac protocol for next generation wireless lans. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 1, pages 1–6 vol.1, Junho 2005.
- [Net] Scalable Networks. Qualnet network simulator. <http://www.scalable-networks.com/>.
- [NHH04] A. Nosratinia, T.E. Hunter, and A. Hedayat. Cooperative communication in wireless networks. *Communications Magazine, IEEE*, 42(10):74–80, Outubro 2004.
- [NKMK04] T. Nagaosa, Y. Kobayashi, K. Mori, and H. Kobayashi. An advanced csma inter-vehicle communication system using packet transmission timing decided

- by the vehicle position. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 111–114, Junho 2004.
- [NTCS99] Sze-Yao Ni, Yu-Chee Tseng, Yuh-Shyan Chen, and Jang-Ping Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 151–162, New York, NY, USA, 1999. ACM.
- [OBP05] Rodolfo Oliveira, Luis Bernardo, and Paulo Pinto. Searching for resources in manets: A cluster based flooding approach. In *ICETE*, pages 105–111, 2005.
- [Oli09] Rodolfo Oliveira. *Controlo de Acesso ao Meio em Redes Ad Hoc Móveis IEEE 802.11*. PhD thesis, 2009.
- [PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, Outubro 1994.
- [PR99] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100, 25-26 Fevereiro 1999.
- [PRL⁺07] Michal Piorkowski, Maxim Raya, Ada L. Lugo, Panos Papadimitratos, Matthias Grossglauser, and Jean-Pierre Hubaux. Trans: Realistic joint traffic and network simulator for vanets. *ACM SIGMOBILE Mobile Computing and Communications Review*, 2007.
- [QLV01] A. Qayyum, A. Laouiti, and L. Viennot. Multipoint relaying: An efficient technique for flooding in mobile wireless networks. In *35th Annual Hawaii International Conference on System Sciences (HICSS'2001)*. IEEE Computer Society, 2001.
- [Ros09] Francisco J. Ros. *UM-OLSR. Software Package retrieved from <http://masimum.inf.um.es/um-olsr/html/>, 2009.*

- [SEE04] R.A. Santos, R.M. Edwards, and A. Edwards. *Cluster-based location routing algorithm for inter-vehicle communication*. In *Vehicular Technology Conference*, 2004. VTC2004-Fall. 2004 IEEE 60th, volume 2, pages 914–918 Vol. 2, Setembro 2004.
- [Sta05] ANSI/IEEE 802.11 Standard. *Ieee 802.11e/d13.0, draft supplement to part 11: Medium access control (mac) quality of service (qos) enhancements*, January 2005.
- [Stu] University Stuttgart. *Canu project*. <http://canu.informatik.uni-stuttgart.de/>.
- [SYGD08] Christoph Sommer, Zheng Yao, Reinhard German, and Falko Dressler. *On the need for bidirectional coupling of road traffic microsimulation and network simulation*. In *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 41–48, New York, NY, USA, 2008. ACM.
- [Tah87] Hamdy A. Taha. *Operations Research: An Introduction*, 4th ed. Macmillan Publishing Co., Inc., Indianapolis, IN, USA, 1987.
- [WER⁺03] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann. *Adaptive broadcast for travel and traffic information distribution based on inter-vehicle communication*. In *Intelligent Vehicles Symposium*, 2003. Proceedings. IEEE, pages 6–11, Junho 2003.
- [WFR04] Hao Wu, R. Fujimoto, and G. Riley. *Analytical models for information propagation in vehicle-to-vehicle networks*. In *Vehicular Technology Conference*, 2004. VTC2004-Fall. 2004 IEEE 60th, volume 6, pages 4548–4552 Vol. 6, Setembro 2004.
- [XSJC03] Qing Xu, R. Segupta, D. Jiang, and D. Chrysler. *Design and analysis of highway safety communication protocol in 5.9 ghz dedicated short range communication spectrum*. In *Vehicular Technology Conference*, 2003. VTC 2003-Spring. The 57th IEEE Semiannual, volume 4, pages 2451–2455 vol.4, Abril 2003.

- [ZZJ07] Jin Zhang, Qian Zhang, and Weijia Jia. *A novel mac protocol for cooperative downloading in vehicular networks*. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE, pages 4974–4978, Novembre 2007*.